RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Mohamed Khider – BISKRA
Faculté des Sciences Exactes et des Sciences de la Nature et de la Vie
Département d'informatique

N° d'ordre :.....

T H È S E

Pour obtenir le titre de
DOCTEUR EN SCIENCES

Spécialité : INFORMATIQUE

Présentée par :
Houcine BELOUAAR

Modélisation d'une approche basée agent et logique floue pour la qualité des services Web

---

Thèse dirigée par :
Pr. Okba KAZAR

Soutenue devant le jury composé de :

Président :	Mohamed BENMOHAMMED	Professeur	Université de Constantine2
Rapporteur :	Okba KAZAR	Professeur	Université de Biskra
Examinateurs :	Saber BENHARZALLAH	M.C.A	Université de Batna2
	Laid KAHLOUL	M.C.A	Université de Biskra
	Sadek Labib TERRISSA	M.C.A	Université de Biskra
	Djamel NESSAH	M.C.A	Université de Khenchela

Année universitaire : **2018 – 2019**

Remerciements

Tout d'abord, Mes louanges à **DIEU** le Tout Puissant pour m'avoir donné le courage, la volonté, la patience durant ces années d'étude et que grâce à Lui ce travail a été réalisé.

Les travaux de recherche dans le cadre de cette thèse sont effectués au sein du Laboratoire de l'INFormatique Intelligente Biskra (L I N F I) de l'université Mohamed Khider sous La direction de Mr **Okba KAZAR** Professeur à l'université Mohamed Khider de Biskra. Qu'il trouve ici le témoignage de ma profonde gratitude et mes sincères remerciements. Ses qualités scientifiques et humaines ont Toujours été une source de motivation. Son aide et ses conseils m'ont été toujours précieux.

Je remercie très vivement Mr **Mohamed BEN MOHAMMED** Professeur à l'université de Constantine2 pour l'honneur qu'il me fait en acceptant de présider le jury.

Mes vifs remerciement vont également aux honorables membres de jury : Mr **Saber BENHARZALLAH** Maître de conférences à l'université de Batna2, Mr **Sadek Labib TERRISSA** Maître de conférences à l'université Mohamed Khider de Biskra, Mr **Laid KAHLOUL** Maître de conférences à l'université Mohamed Khider de Biskra, Mr **Djamel NESSAH** Maître de conférences à l'université de Khenchela.

Un remerciement particulier à mon épouse et mes enfants pour leur patience à mon indisponibilité permanente durant la préparation de cette thèse.

Houcine BELOUAAR

À la mémoire de mes defunts parents.

À ma chère famille.

.....

ملخص:

أصبح اختيار خدمة ويب من بين العديد من الخدمات المماثلة مهمة حساسة للغاية بالنسبة لمستهلك خدمات الويب لأنه يواجه صعوبة في الاختيار ولن يتمكن من تحديد الخدمة الأكثر ملاءمة. يمكن أن يكون الفرق الوحيد بين خدمات الويب المشابهة هو معايير جودة الخدمة (QoS) وتحديدًا خصائصها غير الوظيفية التي لها تأثير كبير على اختيار أفضل خدمة.

من ناحية أخرى، فإن معايير الجودة غير الوظيفية المتعلقة بخدمات الويب للمستهلك غير دقيقة وغير مؤكدة أحيانا وغامضة لأن المعطيات الرقمية غير قادرة في بعض الأحيان على نموذجة مواقف حقيقية وبالتالي المنطق الضبابي يأتي لترجمة هذا التقييم مستعملا معايير جودة الخدمة بطريقة مناسبة للمستهلك.

في هذا العمل ، نقترح نهجًا يعتمد على المنطق الضبابي و تقنية العميل لاختيار خدمات الويب ، وذلك بأخذ بعين الإعتبار المعطيات اللغوية للمستهلك ثم تقديم خدمات الويب المطلوبة. لذلك ، يعد استخدام مفهوم الوكيل لخدمات الويب تحديًا كبيرًا لتزويد خدمات الويب بقدرات مهمة من وكلاء البرامج.

مهمتنا هي إضافة وحدة تعتمد على المنطق الضبابي تقوم بتحويل المعطيات من قيم حقيقية إلى قيم ضبابية بالإضافة إلى التصنيف متعدد المعايير لخدمات الويب المماثلة. بالنسبة لتصنيف خدمات الويب المشابهة ، اخترنا حلّين: الأول هو استخدام محرك الاستدلال الضبابي الذي يعتمد على القواعد الضبابية ؛ ويستند الثاني على استخدام أسلوب ترتيب التفضيل عن طريق التشابه مع الحل المثالي (TOPSIS) كون أننا نواجه مشكلة MCDM .

الكلمات المفتاحية: خدمة ويب ، اختيار ، جودة الخدمة (QoS) ، المنطق الضبابي ، الترتيب ، ضبابي TOPSIS ، استدلال ضبابي ، نظام متعدد العوامل.

Abstract

Selecting a web service from several similar services has become a very delicate task for the service consumer because he is faced with the embarrassment of choice and will be unable to decide which service is most appropriate. The only difference between similar Web services can be their quality of service (QoS) and more specifically their non-functional properties which have a great influence on the selection of the best service.

On the other side, the criteria of quality of services related to the Web services consumers are imprecise and sometimes uncertain and ambiguous because crisp data are inadequate to model real-life situations therefore the fuzzy logic comes to translate this assessment by presenting the criteria of qualities in an adequate way to the consumer.

In this work, we propose an approach based agent and fuzzy logic to select web service, so that to take into account the consumer's linguistic fuzzy data and subsequently render it the desired web services. The use of the notion of agent for web services is thus a major challenge to equip web services with interesting capabilities of software agents.

Our work is to add a fuzzy module that consists of the fuzzification of quantitative criteria as well as the multi-criteria ranking of similar web services. For the classification of similar Web services, we have opted for two solutions : the first is the use of a fuzzy inference engine based on fuzzy rules; the second uses the technique of order of preference by similarity to the ideal solution (TOPSIS) since we are confronted with a problem MCDM.

Keywords : Web service, selection, QoS(Quality of Service), fuzzy logic, ranking, Fuzzy TOPSIS, Fuzzy inference, Multi-Agent System

Résumé

La sélection d'un service Web parmi plusieurs services similaires est devenue une tâche très délicate pour le consommateur de services car il est confronté à l'embarras du choix et il sera incapable de décider quel service est le plus approprié. La seule différence entre des services Web similaires peut être leur qualité de service (QoS) et plus précisément leurs propriétés non fonctionnelles qui ont une grande influence sur la sélection du meilleur service.

D'autre part, les critères de qualité des services liés aux services Web consommateurs sont imprécis et parfois incertains et ambigus car les données quantitatives sont insuffisantes pour modéliser des situations réelles donc la logique floue vient pour traduire cette évaluation en présentant les critères de qualités d'une manière adéquate pour le consommateur.

Dans ce travail, nous proposons une approche basée agent et logique floue de sélection des services Web afin de prendre en compte les données floues linguistiques du consommateur et de le rendre ensuite les services web souhaités. L'utilisation de la notion d'agent pour les services web est donc un défi majeur pour équiper les services web avec des capacités intéressantes d'agents logiciels.

Notre travail consiste en l'ajout d'un module flou qui consiste en la fuzzification des critères quantitatifs ainsi que le classement multi-critères de services web similaires. Pour le classement des services Web similaires, nous avons opté pour deux solutions : la première consiste en l'utilisation d'un moteur d'inférence flou qui repose sur des règles floues ; la seconde utilise la technique de l'ordre de préférence par similitude à la solution idéale (TOPSIS) puisque nous sommes confrontés à un problème MCDM.

Mots clés : Service Web, sélection, qualité de service (QoS), logique floue, classement, TOPSIS flou, inférence floue, système multi-agent.

Table des figures

Liste des tableaux

Liste des abréviations

AI/IA : Artificial Intelligence/Intelligence artificielle

BDI : Belief-Desire-Intention

BPEL : Business Process Execution Language

BPML : Business Process Modeling Language

COG : Centre of gravity

DAML : DARPA Agent Markup Language

DAML-S : DARPA agent markup language for services)

FNIS : Fuzzy Negative Ideal Solution

FPIS : Fuzzy Positive ideal solution

MAS/SMA : Multi agent system/Système multi agents

MCDM : Multiple Criteria Decision Making

MF : Membership function

MoM : Mean of maxima method

OWL-S : Semantic markup for web services

QoS : Quality of service

RDF : Resource Description Framework

SAWSDL : Semantic annotations for WSDL and XML schema

S-CDL : Web Services Choreography Description Language

SLA : Service Level Agreement

SOA : Service-Oriented Architecture

SOAP : Simple Object Access Protocol

TOPSIS : Technique for Order Preference by Similarity to Ideal Solution

UDDI : Universal Description Discovery and Integration

UML : Unified Modeling Language

URI : Uniform Resource Identifier

URL : Uniform Resource Locator

USDL : Unified Service Description Language

W3C : World Wide Web Consortium

WAM : Weighted average method

WSCI : Web Service Choreography Interface

WSDL : Web Services Description Language

WSDL-S : Web Service Description Language Semantic

WSMO : Web service modeling ontology

XML : Extensible Markup Language

Table des matières

Chapitre 1

Introduction Générale

1.1 Contexte du travail

Un *service Web* peut être définie essentiellement comme une abstraction sémantiquement bien définie d'un ensemble d'activités informatiques ou physiques impliquant un certain nombre de ressources, destinées à répondre à un besoin client ou à un besoin métier[1].

Les services Web émergent comme un cadre systématique et extensible pour l'interaction d'application à application, basé sur les protocoles Web existants et les standards XML ouverts. Ils sont une nouvelle génération d'applications Web qui sont des applications modulaires autonomes, auto descriptives, qui peuvent être publiées, localisées et invoquées sur le Web[2]. Les services Web exécutent des fonctions qui peuvent aller de simples demandes d'informations à la création et à l'exécution de processus métier complexes. Une fois qu'un service Web est déployé, il peut être découvert et appelé par d'autres applications.

Avec la popularité de l'internet, les services Web sont devenus indispensables dans presque tous les domaines. Ils constituent une catégorie de solutions Web utilisées dans les industries, telles que les services bancaires, shopping, et les communications. Les services Web peuvent être décrits simplement comme n'importe quel service offert sur le Web. Par exemple, dans une application Web météo, un utilisateur ouvre un site Web météorologique et saisit des données (par exemple, code postal), le serveur de site traite les données et envoie la réponse, dans ce cas, une prévision météorologique.

Les services Web reposent sur une technologie permettant à des applications de dialoguer à

distance via Internet, et ceci indépendamment des plates-formes et des langages sur lesquelles elles reposent. Ils se sont grandement utilises par les entreprises, ce qui leur permet d'exposer un certain nombre de services et d'échanger les informations entre elles.

Les services Web constituent la technologie la plus adaptée à l'architecture orientée service(Service oriented architecture) qui est est un style architectural pour la construction de solutions d'entreprise basées sur les services[4]. l'architecture Service oriented architecture est généralement divisée en trois couches[6] : Fournisseur de services : Le prestataire de services développe les services et les rend accessibles sur Internet pour les utilisateurs. Demandeur de service : le demandeur de service est un utilisateur final du service Web. Le demandeur consomme les services Web déjà accessibles renvoyés par le fournisseur de services. L'annuaire des services est un emplacement centralisé pour tous les services Web.

Les services Web s'appuient sur un ensemble de protocoles Internet très répandus, afin de communiquer. Cette communication est basée sur le principe de demandes et réponses, effectuées avec des messages XML : L'architecture traditionnelle des services Web est basée sur le modèle SOA, qui implique trois acteurs : l'utilisateur ou client ou du service, le fournisseur de services et un intermédiaire jouant le rôle Registre des services.

1. Le fournisseur du service : Cela correspond à la personne ou à l'organisation propriétaire du service qui effectuera le service demandé. Il publie son service en fournissant une description des services au format WSDL dans le répertoire de service afin que les clients puissent découvrir et accéder au service. IL représente donc l'environnement d'hébergement et d'exécution du service.
2. L'annuaire du registre : C'est un registre descriptif des services Web qui fournit aux fournisseurs les moyens de publier et d'indexer leurs services Web sur le réseau. Il permet également aux clients de rechercher, localiser et trouver ces services selon plusieurs critères répondant à leurs besoins. En d'autres termes, le répertoire agit en tant qu'intermédiaire entre les clients et les fournisseurs de services.
3. Le client du service : Il représente tout consommateur du service Web. D'un point de vue technique, le demandeur de service est constitué de l'application qui recherche et invoque un service en envoyant une requête.

Les services Web reposent sur quatre technologies de base[2][3] :

1. le protocole SOAP (Simple Object Access Protocol) permettant la communication entre services Web ;
2. Le répertoire UDDI (Universal Description, Discovery and Integration) qui est un registre des descriptions de services Web ;
3. Le répertoire UDDI (Universal Description, Discovery and Integration) qui est un registre des descriptions de services Web.
4. Markup Language (XML) : XML a été créé comme un moyen auto-descriptif de représenter des données totalement indépendantes de l'application, du protocole, du vocabulaire, du système d'exploitation ou même du langage de programmation.

Les services ont été beaucoup utilisés grâce aux avantages qu'ils présentent :

— Les services Web sont très appropriés pour intégrer des systèmes informatiques complètement différents les uns des autres. Cet avantage est l'un des plus grands appels pour les grandes compagines informatiques telles que IBM et Microsoft pour améliorer l'e-business.

— Les services Web sont très rapides et peu coûteux à développer. De nombreux outils sont disponibles qui utilisent une certaine description d'un service Web, spécifiée dans le langage de description de service Web (wsdl). En utilisant une telle description, ces outils génèrent du code presque compilable dans un certain langage de programmation[5].

— Les services Web sont utiles. Il existe différents conteneurs pour différents protocoles de couche de transport tels que : FTP, SMTP et HTTP pour déployer rapidement un service Web.

— Les services Web sont faciles à découvrir et à invoquer.

Le contexte de notre travail tourne autour des concepts suivants :

— *Service Web :* Le développement de systèmes distribués[7] est influencé par plusieurs paradigmes. En outre, des technologies telles que les services Web sont maintenant considérés comme la norme, déployée dans les outils de développement communs et largement utilisé.

— *Qualité de service :* La qualité de service[8] peut être définie comme une comparaison entre les attentes du client et ce qui est réellement offert. Elle est défini comme une combinaison de plusieurs critères qui peuvent alors être considérés comme un critère

de choix lorsque l'on sélectionne parmi plusieurs services web découverts ceux qui respectent les contraintes imposées.

— *Critères de qualité de service :* Plusieurs critères de qualité de services sont décrits dans différents travaux et les propriétés de QoS les plus représentatives sont présentées comme suit : Availability, Reliability, Throughput, Response time, Accessibility, Execution, price, Reputation...etc.

— *Logique floue :* La qualité des modèles de services [8] ont tendance à conduire à une représentation pauvre de la réalité, principalement en raison de leur manque de capacité à représenter l'imprécision. La logique floue fournit[9] un cadre naturel pour faire face à l'incertitude et la tolérance aux données imprécises.

— *Agent :* Les agents logiciels sont envisagés comme des entités logicielles autonomes dynamiques qui agissent pour le compte de ses utilisateurs en fonction d'une donnée d'ordre du jour de buts. L'utilisation de la notion d'agent pour les services web est ainsi un enjeu de taille pour doter les services web des capacités intéressante des agents logiciels.

— *Sélection de service web :* Lors de la recherche de services Web de base, certains travaux intègrent le contexte d'utilisation pour trouver les services Web les plus appropriés à la demande de l'utilisateur. Le système peut disposer d'un ensemble de services Web pouvant répondre fonctionnellement à la demande. La technique de sélection implique alors de sélectionner le service Web le mieux adapté au contexte.

Une technique de sélection consiste en un ensemble d'instructions pour un service sur le Web au moment de l'exécution, sur la base des informations disponibles dans le système. Deux cas de sélection peuvent être existés : la sélection de base des éléments fonctionnels et la sélection basée sur les besoins non fonctionnels.

1.2 Problématique et objectifs

En raison du grand succès des services Web grâce aux technologies qui favorisent l'interopérabilité, l'extensibilité, l'indépendance vis-à-vis des plates-formes et des langages de programmation, leur utilisation est devenue de plus en plus indispensable ainsi le nombre des services Web déployés dans le Web est en croissance sans cesse. Une simple requête d'un client peut

avoir des centaines de réponses voir des milliers de ce que fait, le client tombe dans l'embarras de choix et se trouve face à un problème de choix entre les services similaires retournés par la requête car les services fournissant la fonctionnalité en question seront retournés dans l'ordre de leur inscription dans le registre UDDI en question[10]. Tout fois, il est très rare de répondre à une requête client par un seul service.

Généralement, un processus de découverte de service Web s'effectue en trois étapes : La première étape est la publication du service Web par les fournisseurs qui annoncent les services Web dans l'annuaire en enregistrant leurs services Web à l'aide du fichier de description du service Web écrit en WSDL. La deuxième étape est la demande de service Web par l'utilisateur. L'utilisateur envoie une demande de service Web spécifiant l'exigence dans un format prédéfini à l'annuaire de service Web. Le service Web Matching fait correspondre la requête de l'utilisateur avec les services Web disponibles et trouve un ensemble de candidats au service Web.

Les approches traditionnelles de découverte de service Web est une recherche basée sur des mots dans le registre UDDI. Diverses autres approches pour découvrir des services Web sont également disponibles. Certaines des approches de découverte sont basées sur la syntaxe tandis que d'autres sont basées sur la sémantique[11].

Donc, il est important de trouver un moyen pour répondre au client en lui servant le service le plus adéquat. Plusieurs solutions ont été proposées dans ce sens, les meilleures solutions peuvent être celles qui s'appuient sur les critères de qualité de services(QoS) et plus particulièrement sur les paramètres de critères non fonctionnelles[66][20]. De cela, un service Web se compose d'un aspect fonctionnelle qui decrit ce que le service doit faire et un aspect non fonctionnel qui decrit comment le service peut le faire. En sélectionnant un service Web à base de QoS, revient à résoudre un problème multicritères(MCDM [1]).

En plus de cela, dans de nombreux cas, la valeur de la propriété QoS peut être difficile à définir avec précision car les qualités des services liés aux consommateurs de services Web sont imprécises et parfois incertaines et ambiguës en raison de l'état mental et du manque d'information des consommateurs sur le contenu des services Web et aussi sur leur QoS. Encore, les valeurs et les pondérations des attributs QoS (notamment QoS qualitative) ne sont pas faciles à définir avec précision, donc, il est donc préférable que les préférences utilisateur uti-

1. Multiple Criteria Decision Making

lisées comme propriétés QoS soient floues car mieux adaptées à l'interprétation des termes linguistiques[67]. Par conséquent, la logique floue peut être appliquée pour soutenir la représentation de contraintes QoS imprécises[65].

Le développement de systèmes distribués[7] est influencé par plusieurs paradigmes. En outre, des technologies telles que les services Web sont maintenant considérés comme la norme, déployée dans les outils de développement communs et largement utilisé. Cependant, malgré cette tendance récente, le nombre sans cesse croissant de puissants dispositifs personnels va inévitablement relancer l'intérêt dans un autre paradigme connu sous le nom d'agents autonomes. Les agents sont en effet considérés comme l'un des principaux éléments constitutifs de l'infrastructure Web de la prochaine génération émergente. Les services Web sont des ressources très importantes pour les agents. Les agents doivent être en mesure de récupérer, d'exécuter et de composer des services Web, fournissant un support intelligent et personnalisé pour les utilisateurs. D'autre part, les agents doivent également être en mesure d'exporter leurs fonctionnalités comme les services Web afin d'être pleinement intégrés dans le paradigme orienté Service.

L'objectif de cette thèse est de proposer une approche basée agent et logique floue(les raisons sont expliquées dans les deux paragraphes précédentes) pour la qualité des services Web. Cette approche doit d'une part permettre aux utilisateurs des services Web de lancer leurs requêtes en termes linguistiques et d'autre part, présenter les différents paramètres de qualité de service en valeurs floues permettant à la fin de retourner aux clients les services les plus adéquats.

1.3 Contributions

Comme solutions à la proplématique décrite précédemment, nous proposons deux contributions :

La première contribution consiste en la proposition d'un modèle basé logique floue pour la sélection des services Web. Cette contribution a été enrichie avec un algorithme de sélection des services Web qui décrit le processus de sélection floue, dès l'émission de la requête par l'utilisateur, passant par le sous-système flou qui s'occupe de la transformation floue et de l'inférence

jusqu'à le retour du meilleur service au demandeur.

La deuxième contribution, est une approche de sélection des services Web basée sur la notion d'agent et de la logique floue. Elle est constituée principalement d'une couche floue jouant le rôle d'un sous-système flou qui s'occupe de la transformation des critères de qualité de services en valeurs floues, formant une matrice de décision floue qui sera transmise l'agent de classement floue, qui utilise l'algorithme TOPSIS [1] flou. Pour classer la liste des services Web en fonction des critères afin de fournir le meilleur service qui sera envoyé à l'agent Client qui est ensuite mis à disposition au client final pour qu'il puisse invoquer le service.

1.4 Structure de la thèse

Après une conclusion générale decrivant le contexte du travail, la problématique et contribution, la présente thèse est organisée en comme suit :

Dans sa première partie, le deuxième chapitre décrit l'état de l'art destiné aux technologies des services Web. Le troisième chapitre et le quatrième chapitre se rapportent à l'état de l'art qui concerne respectivement la logique floue et l'aspect agent. Dans le cinquième chapitre, nous présentons quelques travaux reliés à notre problématique.

La deuxième partie se constitue de la contributions et des résultats expérimentataux; Nous cloturons cette partie avec une conclusion générale et quelques perspectives futures.

1. The Technique for Order of Preference by Similarity to Ideal Solution

Première partie

Etat de l’art

Chapitre 2

Technologies des services Web

2.1 Introduction

Avec la technologie des services Web, les entreprises peuvent exposer leurs processus métier internes en tant que services et les rendre accessibles via Internet ainsi les services Web sont devenus indispensables dans presque tous les domaines. Ces services Web constituent une catégorie de solutions Web utilisées dans les industries, telles que les services bancaires, shopping, et les communications. Ils peuvent être décrits simplement comme n'importe quel service offert sur le Web. Par exemple, un client demande le prix d'un article en envoyant un message sur le web. Ce message contient la référence de l'article. Le service Web va recevoir la référence, effectuer le traitement du service et renvoyer le prix au client via un autre message. En effet, la technologie des services Web est le choix le plus prometteur pour mettre en œuvre une architecture orientée services et ses objectifs stratégiques.

Dans ce chapitre, nous présentons un survol sur cette technologie en commençant l'architecture orientée service puis nous présentons la technologies des services Web dont laquelle, nous exposons les différentes architecture à savoir l'architecture de références et l'architecture en couches et enfin nous terminons ce chapitre par la composition des services Web.

2.2 Architecture orientée Service :

L'Architecture Orientée Service (SOA [1]) est décrite comme une stratégie pour créer des applications orientées service. Son objectif est de fournir des services pouvant être utilisés par d'autres services[12]. L'architecture de référence des Services Web est l'architecture SOA ou l'architecture orientée service qui est née principalement pour répondre aux inconvénients des technologies orientées composants tels que : la complexité, le manque de compatibilité puisque ces composants étaient difficilement interopérables entre eux et la modification dans un composant conduit généralement à une défaillance du système.

2.3 Définition de l'architecture orientée Service

Dans la littérature, il y a plusieurs définition de l'architecture orientée Service. Voici quelques définitions de l'architecture SOA :

« Une architecture orientée services(SOA) est essentiellement une collection de services. Ces services communiquent entre eux. La communication peut impliquer soit un simple transfert de données, soit deux ou plusieurs services coordonnant une activité. Certains moyens de connecter les services les uns aux autres sont nécessaires[13]. »

« Une architectures orientées services (SOA)est un paradigme qui permet l'intégration d'applications et de ressources de manière flexible en :

— Représentant chaque application ou ressource sous la forme d'un service exposant une interface standardisée ;
— Permettant à un service d'échanger des informations structurées (messages, documents, objets métier) ;
— Coordonnant et organisant les services afin d'assurer qu'ils puissent être invoqués et utilisée de manière efficace [14]. »

« une architectures orientées services (SOA) est un paradigme récent pour la construction d'applications logicielles à grande échelle à partir de services distribués [15]. L'un des principaux intérêts de SOA est fondamentalement la capacité sous-jacente d'une telle architecture à être

1. Service-Oriented Architecture

intrinsèquement évolutive ; car l'idée sous-jacente de SOA est que les services sont faiblement couplés et que la SOA peut être adaptée à son environnement. »

« L'architecture orientée services est un modèle utilisé pour transformer les composants d'un système d'information en services qui peuvent être intégrés pour construire des processus inter-métiers. Les services sont fournis à d'autres composants via un protocole de communication, généralement sur un réseau. Les principes d'orientation de service sont indépendants de tout fournisseur, produit ou technologie[16]. »

Le plus important dans l'Architecture SOA est la normalisation. Il définit les spécifications de tous les types d'application à découvrir et à communiquer les uns avec les autres. D'une part, cela permet également de faire usage de systèmes existants pour coopérer avec les nouvelles technologies modernes. D'autre part, la description de l'interface standard permet d'implémenter le service avec différents langages, plates-formes, modèles d'objets ou systèmes de messagerie.

2.3.1 Avantages de l'architecture Orientée Service

L'architecture SOA regroupe beaucoup d'avantages, parmi ces avantages, nous pouvons citer :

— Couplage faible entre les services ;
— Sécuriser l'investissement des applications existantes
— L'indépendance par rapport aux aspects technologiques ;
— Architecture basée sur des standards
— Réduire les coûts et le temps de développement en réutilisant les actifs existants ;
— Facilité de maintenance et amélioration de la flexibilité et de l'évolutivité

D'après ces définitions de nous pouvons conclure que cette architecture repose principalement sur l'aspect service Web, d'où, une architecture orientée service est un paradigme fondée sur la description et l'interaction de services. Tout simplement, un service web est le fait de mettre des ressources à disposition (gratuite ou non) sur Internet, via un protocole d'échanges standardisé, pour des programmes écrits dans des langages quelconques.

En bref, l'architecture orientée services est un paradigme permettant d'organiser et d'utiliser des savoir-faire distribués pouvant être de domaines variés. Cela fournit un moyen uniforme

d'offrir, de découvrir, d'interagir et d'utiliser des savoir-faire pour produire le résultat désiré avec des pré-conditions et des buts mesurables.

À partir de ces définitions, nous ressortons une idée principale, qui malgré le manque de spécification officielle pour définir une architecture orientée service, trois rôles clés sont communément identifiés :

— fournisseur de services ;

— registre de service

— demandeur de service.

Le fournisseur de services a pour fonction de déployer un service sur un serveur et de générer une description de ce service. Cette dernière précise à la fois les opérations disponibles et leur mode d'invocation. Cette description est publiée dans un registre de service ou annuaire. Les consommateurs peuvent découvrir les services disponibles et obtenir leur description en lançant une recherche sur un répertoire. Ils peuvent ensuite utiliser la description du service ainsi obtenue pour établir une connexion avec le fournisseur et invoquer les opérations du service souhaité.

L'approche la plus répandue pour mettre en œuvre une architecture SOA est l'utilisation des services Web. Cette technologie vise à la transposition des SOA dans le cadre du Web.

2.4 Service Web

D'une manière générale, on peut définir un service web comme étant toute application accessible à d'autres applications sur le Web. C'est une définition très ouverte, dans laquelle presque tout ce qui a un URL (Universal Resource Locator) est un service Web[1].

2.4.1 Définition des services Web

Plusieurs définitions définitions des services web ont été proposées. Nous citons celles de W3C, d'IBM et de Wikipédia.

« Selon Wikipedia, Un service web est un protocole d'interface informatique de la famille des technologies web permettant la communication et l'échange de données entre applications et systèmes hétérogènes dans des environnements distribués. Il s'agit donc d'un ensemble de

fonctionnalités exposées sur internet ou sur un intranet, par et pour des applications ou machines, sans intervention humaine, de manière synchrone ou asynchrone. »

« Selon W3C, Un service Web est un système logiciel identifié par un URI, dont les interfaces publiques et les liaisons sont définies et décrites à l'aide de XML. Sa définition peut être découverte par d'autres services web. Ces systèmes peuvent ensuite interagir avec le service Web d'une manière prescrite par sa définition, en utilisant des messages XML en utilisant des protocoles internet. »

« Selon IBM, Un service Web est un terme générique désignant une fonction logicielle interopérable de machine à machine hébergée dans un emplacement adressable par le réseau. Il dispose d'une interface qui masque les détails d'implémentation de façon à pouvoir être utilisé indépendamment de la plate-forme matérielle ou logicielle sur laquelle il est implémenté et indépendamment du langage de programmation dans lequel il est écrit[111]. Cette indépendance encourage les applications basées sur le service Web à être faiblement couplées, orientées composants, implémentations inter-technologies. Les services Web peuvent être utilisés seuls ou avec d'autres services Web pour effectuer une agrégation complexe ou une transaction commerciale.

« Un service est une fonction logicielle autonome et sans état qui accepte des requêtes et qui renvoie des réponses au travers d'une interface standard bien défini[114]. »

Un service est donc une unité de traitement qui fournit un résultat à un consommateur. Fournisseurs et consommateurs sont habituellement des agents logiciels qui agissent par délégation de leurs propriétaires.

2.5 Technologies des services Web :

WSDL [1], SOAP [2] et UDDI [3] constituent l'ensemble des technologies clés des services Web, sur lesquelles d'autres technologies plus proches de la problématique applicative peuvent être spécifiées et mises en œuvre [12] :

— Simple Object Access Protocol (SOAP) : SOAP est un protocole léger destiné à échan-

1. Web Service Description Language
2. Simple Object Access Protocol
3. Universal Description Discovery and Integration

ger des informations structurées dans un environnement décentralisé et distribué. Il utilise des technologies XML pour définir un cadre de messagerie extensible fournissant une construction de message pouvant être échangée sur une variété de protocoles sous-jacents[113]. Le cadre a été conçu pour être indépendant de tout modèle de programmation particulier et de toute autre sémantique spécifique à l'implémentation. L'enregistrement du service, la découverte et l'invocation sont implémentés par les appels SOAP.

— WSDL (Web Service Description Language) : W3C définit WSDL comme "un format XML pour décrire les services réseau comme un ensemble de points de terminaison fonctionnant sur des messages contenant des informations orientées document ou orientées procédure"[112]. Il décrit les opérations qui composent un service, les messages échangés par chaque opération, les parties qui forment chaque message et les liaisons de protocole pour interagir avec le service dans les normes prédéfinies [19].

— Découverte et intégration de descriptions universelles (UDDI) : UDDI est défini comme : un ensemble de services supportant la description et la découverte d'entreprises, d'organisations et d'autres fournisseurs de services Web, les services Web qu'ils mettent à disposition et les interfaces techniques qui peuvent être utilisées pour accéder à ces services par OASIS (Organisation pour l'avancement des normes d'information structurées) [17]. UDDI est une initiative de l'industrie qui permet aux entreprises de publier leurs services et permet aux utilisateurs potentiels de découvrir ces services. Les registres UDDI peuvent être publics ou privés et les utilisateurs peuvent rechercher et sélectionner un service approprié à partir d'un registre UDDI.

2.5.1 Propriétés d'un service Web

Les propriétés que possède un service Web sont nombreuses et assez diversifiées selon l'aspect que couvre le service Web.

Elles peuvent aller d'une simple information sur ce que fait un service Web jusqu'au détail sur la manière de le faire et avec quelle garantie du point de vue sécurité ou qualité de service en passant par les détails techniques nécessaires pour l'accès réel au service.

On distingue les propriétés suivantes : fonctionnelles, non-fonctionnelles, contextuelles, orientées données, techniques et enfin comportementales [18] :

Propriétés fonctionnelles

Une propriété fonctionnelle d'un service Web est une propriété relative à la fonctionnalité du service Web. Dans ce cas, les services sont considérés comme une fonction ayant des paramètres, les propriétés fonctionnelles sont liées au type de service, au nom de l'opération, au format et à la sémantique des données d'entrée/sortie.

Les propriétés fonctionnelles comprennent un ensemble de propriétés qui constituent une spécification de ce que peut offrir un service Web à ses clients en termes de fonctionnalités, lorsqu'il est invoqué.

Propriétés non-fonctionnelles

les propriétés non-fonctionnelles comprennent un ensemble de propriétés qui décrivent l'aspect non-fonctionnel du service Web. Ces propriétés sont souvent paramètres des services Web qui n'informent pas sur son aspect fonctionnel. Elles constituent en fait des paramètres qui régissent son comportement ou son utilisation et elles sont généralement exprimées en utilisant des critères de qualité de service tels que la fiabilité, la disponibilité, le temps de réponse...etc [20].

Propriétés contextuelles

Les proprriétés contextuelles d'un service sont des propriétés qui sont en relation avec le contexte de leurs propriétés fonctionnelles et non-fonctionnelles. Il relève de leur contexte d'utilisation et comprend des propriétés telles que l'unité et l'échelle qui doivent être associées à des valeurs de propriétés fonctionnelles ou non-fonctionnelles. Si nous considérons à titre d'exemple le cas d'un service Web qui exécute une opération ayant comme entrée le prix d'un article. Des attributs d'ordre contextuel, tels que la devise et le facteur multiplicateur associés à l'entrée « Prix », demeurent nécessaires pour interpréter correctement les données échangées avec le service Web en question[18].

Propriétés orientées données

ce type de proproétés comprend les types des données utilisées par le service Web, ainsi que toute information relative à sa mise en correspondance avec un autre type de données [18]. Les propriétés orientées données permettent d'une part, de pouvoir communiquer correctement avec le service en utilisant des données compatibles avec les structures qu'il utilise, et d'autre part, d'être averti quand il est question de composer le service avec un autre qui n'utilise pas les mêmes types de données et qui nécessite par la suite de réaliser une mise en correspondance entre des types de données hétérogènes.

Propriétés techniques

Les propriétés techniques sont des propriétés qui informent sur l'aspect technique du service Web. Ces propriétés sont indispensables pour le client puisque d'une part, elles lui permettent de savoir accéder au service, et d'autre part, de pouvoir communiquer correctement avec ce service en échangeant des messages dont la structure et les protocoles sont conformes à ce qui est proposé par ce même service[18]

Propriétés comportementales

Les propriétés comportementales sont des propriétés qui renseignent sur le comportement du service Web qui est particulièrement essentiel dans le cadre de la composition des services Web[18]. Elles fournissent des détails sur un aspect important de ce service et doivent nécessairement être mentionnées dans sa description. Ces informations sur le comportement interne et externe du service composite permettent d'une part, d'exécuter dans l'ordre prévu ses services internes quand ce service Web est invoqué, et d'autre part, d'interagir correctement avec des services externes à ce service composite.

2.6 Architecture des services Web

Dans la littérature, il existe deux types d'architectures des services web : la première dite de référence, elle tourne autour de trois éléments principaux : le fournisseur, le registre et le four-

nisseur. La seconde est dite étendue ou en couches qui s'articule sur l'architecture de référence et ajoutant d'autres couches plus spécifiques.

2.6.1 Architecture de référence de service Web

L'architecture de référence s'articule sur trois points essentiels[21] :

Le fournisseur de services, l'annuaire du registre et le consommateur de service. La figure suivante montre une représentation graphique du modèle de service Web traditionnel :

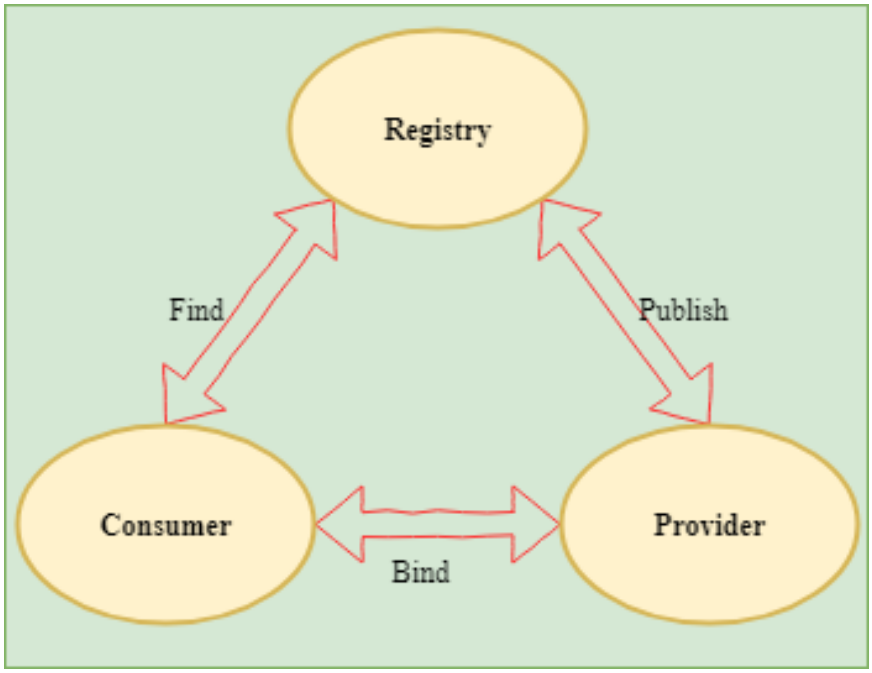

FIGURE 2.1 – Architecture de référence des Services Web[22]

Le fournisseur de services crée ou offre simplement le service Web. Le fournisseur de services doit décrire le service Web dans un format standard XML et le publier dans un registre de service central.

L'annuaire du registre contient des informations supplémentaires sur le fournisseur de services, telles que l'adresse et le contact de la société fournissant, et des détails techniques sur le service.

Le consommateur de service extrait les informations du registre et utilise la description de service obtenue pour lier et invoquer le service Web.

L'architecture des services Web est faiblement couplée, axée sur le service. Le langage de description de service Web WSDL utilise le format XML pour décrire les méthodes fournies par un service Web, y compris les paramètres d'entrée et de sortie, les types de données et le protocole de transport, généralement HTTP, à utiliser. La norme de découverte et d'intégration de la description universelle UDDI suggère de publier des détails sur un fournisseur de services, les services qui sont stockés et la possibilité pour les consommateurs de services de trouver des fournisseurs de services et des détails de services Web.

Ces trois composants interagissent entre eux selon trois types d'opérations : la publication, la recherche et le lien.

En détail, ces opérations sont[23] :

la publication. Pour être accessible, une description de service doit être publiée afin que le demandeur de service peut le trouver. Où il est publié peut varier en fonction de des exigences de l'application.

la recherche. Dans l'opération de recherche, le demandeur de service récupère directement une description de service ou interroge le registre de services pour le type de service requis. L'opération de recherche peut être impliquée dans deux phases de cycle de vie différentes pour le demandeur de service : au moment du design pour extraire la description de l'interface du service pour le développement du programme et au moment de l'exécution pour extraire la description du service et de l'emplacement.

Le lien. Finalement, un service doit être appelé. Dans l'opération de liaison, le demandeur de service appelle ou initie une interaction avec le service au moment de l'exécution en utilisant les détails de liaison dans la description de service pour localiser, contacter et invoquer le service.

2.6.2 Architecture des Services Web en couches

Pour effectuer les trois opérations de publication, de recherche et de liaison d'une manière interopérable, il doit y avoir une pile de services Web qui englobe les normes à chaque niveau. La figure 2.2 montre une pile de services Web conceptuelle. Les couches supérieures s'appuient sur les capacités fournies par les couches inférieures. Les colonnes verticales représentent des exigences qui doivent être traitées à tous les niveaux de la pile. Le texte sur la gauche représente

les technologies standards qui s'appliquent à cette couche de la pile.

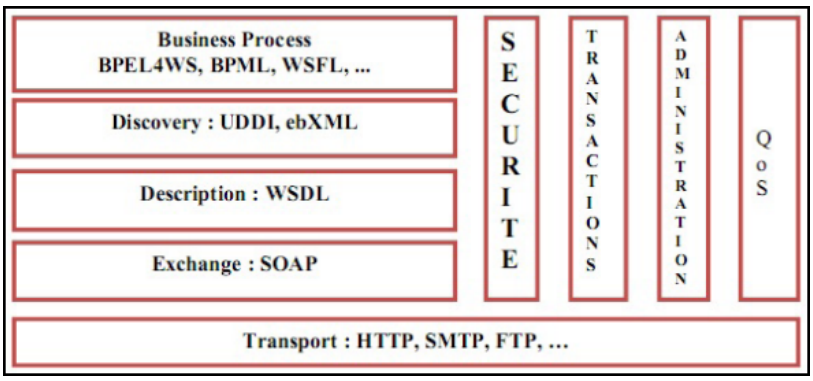

FIGURE 2.2 – architecture en pile des SW[24]

Couche de transport :

Elle définit la structure des messages utilisés par les applications pour se découvrir et dialoguer entre elles. Le protocole de transport le plus couramment utilisé est le protocole HTTP.

Cette couche est à l'heure actuelle la seule réellement normalisée et qui ne souffre d'aucune contestation. Elle s'appuie sur le protocole SOAP (Simple Object Access Protocol) pour l'échange des messages et sur le langage WSDL pour la définition du contrat de l'interface.

Les interactions entre services Web s'effectuent par le biais d'envois de messages structurés au format XML. Le protocole SOAP fournit le cadre permettant ces échanges, il est originellement issu de tentatives précédentes visant à standardiser l'appel de procédures à distance, et en particulier de XML-RPC.

Couche de communication :

Cette couche spécifie les protocoles d'échanges de documents XML entre le service web et ses clients. L'interaction des services Web est basée sur l'échange de messages. Afin d'avoir une compréhension commune de base du contenu de ces messages, le Consortium W3C développe la norme SOAP qui est un protocole basé sur XML pour spécifier comment formater et empaqueter les informations contenues dans ces messages, et comment les transmettre via le réseau.

Un message SOAP (Figure 2.3) est spécifié par une enveloppe composée d'un en-tête SOAP et d'un corps SOAP. Cette structure suit l'approche des protocoles de communication standard

[25] :

— L'en-tête contient les données nécessaires pour le traitement intermédiaire du message ou des protocoles d'infrastructure (comme la sécurité ou la transaction) ;

— Les données d'application que l'émetteur souhaite transmettre au récepteur sont placées dans la partie corps de l'enveloppe SOAP.

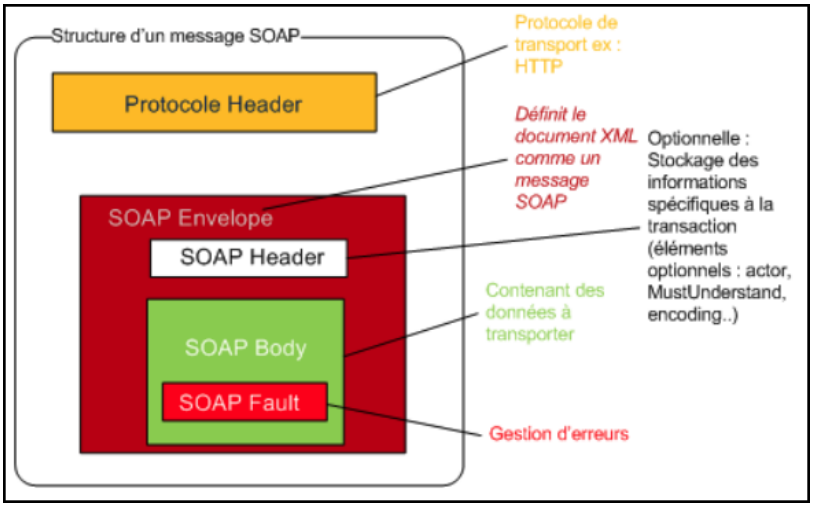

FIGURE 2.3 – Enveloppe SOAP[115]

2.6.3 Couche description :

Le rôle de cette couche est d'assurer une description du service web. Elle concerne l'aspect fonctionnel/non fonctionnel du service web. Les services Web peuvent être syntaxiques, sémantiques, atomiques ou composites[116]. Cette couche est responsable du formatage des données échangées de sorte que les messages peuvent être compris à chaque extrémité. Elle représente un moyen de spécifier une interface publique pour un service web. Elle contient des informations sur les fonctions disponibles, sur les types de données pour les messages XML, des informations de liaison sur le protocole de transport utilisé et l'emplacement d'un service web donné.

Toutes les applications client qui recherchent des informations sur un service utilisent le langage WSDL; elles peuvent ainsi savoir quelles données ce service peut recevoir, s'il renvoie ou non des résultats et le protocole de transport géré. Quand vous créez un service Web, il doit être décrit et rendu public auprès de ses clients potentiels avant de pouvoir être utilisé.

WSDL établit un format commun de description et de publication des informations sur un service web. Généralement WSDL est utilisé avec SOAP, et la spécification WSDL contient une liaison avec SOAP.

Les langages de description des services Web

Il existe plusieurs travaux proposés pour décrire les services web. Ces modèles sont utilisés pour le but de présenter les services de façon compréhensible aux systèmes utilisés. On distingue deux catégories de modèle de description de services qui sont : description à base syntaxique et sémantique. Dans cette section nous allons focaliser sur quelques modèles :

1. WSDL (Web Service Description Language)

   WSDL est un standard du W3C qui permet de définir la structure abstraite et concrète d'un service [26]. Dans le langage WSDL, un service est vu comme une collection de messages pour les échanges et d'une collection de points d'entrée. Un point d'entrée consiste en la description abstraite d'une interface et de son implantation.

   C'est un document XML qui contient le nom de l'opération, le type des paramètres d'entrées/sorties, aussi, il définit des liaisons concrètes pour ces opérations telles que le protocole de transport, l'URI du service, le style du service, et les règles d'encodage employées pour les paramètres d'entrées/sorties [27].

```
<message name="getTermRequest">
  <part name="term" type="xs:string"/>
</message>

<message name="getTermResponse">
  <part name="value" type="xs:string"/>
</message>

<portType name="glossaryTerms">
  <operation name="getTerm">
    <input message="getTermRequest"/>
    <output message="getTermResponse"/>
  </operation>
</portType>
```

FIGURE 2.4 – Exemple WSDL

2. USDL(Unified Service Description Language)

   USDL s'appuie les techniques de description informatique des services tels que WSDL. Cependant, USDL ajoute des informations commerciales et opérationnelles. Les descriptions de service incluent des informations telles que le prix, le service juridique, le fournisseur de services, les méthodes d'interaction, les accords de niveau de service et ainsi de suite[28].

   Pour ce faire, USDL définit des modules UML normatifs pour capturer les données de base d'un service. Cela inclut des modules normatifs, c'est-à-dire des modèles de classe pour la tarification, juridique, les aspects fonctionnels, participants, interaction et SLA. Par conséquent, les services manuels et informatiques peuvent être décrits avec USDL[29].

3. RDF (Ressource Description Framework)

   La description des ressources (RDF; Ressource Description Framework) est proposée comme méthode pour la description conceptuelle ou la modélisation des services Web à travers des graphes basés sur le modèles de données [30]. En général, RDF est une méthode pour décomposer n'importe quel type de connaissance en petit morceaux dont Le graphe d'un RDF contient trois composants principaux qui sont : un objet, un attribut et la valeur[31].

4. DAML-S (DARPA agent markup language for services)

   C'est un langage de balisage d'agent basé sur RDF qui permet la description des services web en utilisant l'aspect sémantique à travers des ontologies. DAML-S offre des moyens afin de créer une description d'un service web qui va être interprété par un programme. En plus, il offre la possibilité de faciliter la description sémantique de services, leurs interfaces et leurs comportements [32].

5. OWL-S (semantic markup for web services)

   OWL-S est une ontologie, dans le cadre OWL du Web Sémantique, pour décrire les Services Web Sémantiques. Il permettra aux utilisateurs et aux agents logiciels de découvrir, d'invoquer, de composer et de surveiller automatiquement les ressources Web offrant des services, sous certaines contraintes. OWL-S est une ontologie OWL particulière. Il fournit aux fournisseurs de services Web un ensemble essentiel de constructions de lan-

gage de balisage pour décrire les propriétés et les capacités de leurs services Web sous une forme non-ambiguë et intelligente [33]. L'automatisation des tâches du service Web, y compris la détection, l'exécution, l'interopérabilité, la composition et la surveillance de l'exécution automatisée du service Web sont rendus facile par le balisage OWL-S [34].

6. WSMO(Web service modeling ontology)

   Web Service modeling ontology (WSMO) vise à décrire tous les aspects pertinents liés aux services via une interface de service Web dans le but ultime d'automatiser les tâches telles que la découverte, la sélection, la composition, médiation, exécution, suivi, etc.) impliqués dans l'intégration intra- et inter-entreprises de services Web[35].

   WSMO identifie quatre éléments de haut niveau (les ontologies, les services Web, les buts et les médiateurs.) comme les principaux concepts qui doivent être décrits afin de définir les services web sémantiques :

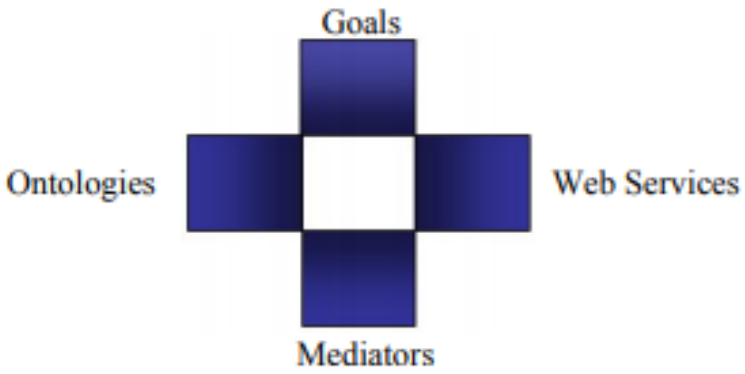

FIGURE 2.5 – Exemple de base WSMO[35]

.

7. WSDL-S (Web Service Description Language Semantic)

   Cette technique prescrit un mécanisme d'annotation des services Web avec la sémantique. WSDL-S a permis d'augmenter l'expressivité de WSDL avec la sémantique en utilisant des concepts similaires à ceux de OWL-S. L'avantage pour cela est double d'une part, en ajoutant la sémantique à WSDL en plusieurs fois, décrivant à la fois la sémantique et les opérations et les détails de niveau dans WSDL, un langage que la communauté des développeurs connaît bien, d'autres parts, en externalisant les modèles de domaine

sémantique[36].

8. SAWSDL (semantic annotations for WSDL and XML schema)

   SAWSDL est un langage sémantique de description de service Web définit comme un standard pour spécifier comment les liaisons de données du service Web peuvent être mises en correspondance avec les modèles formels. Il est évolutif et compatible avec les standards des services Web existants, et plus spécifiquement avec WSDL. De plus, SAWSDL fournit un mécanisme permettant d'annoter sémantiquement les types de données, les opérations, les entrées/sorties de WSDL[37].

2.6.4 Couche découverte de service

Cette couche est chargée de centraliser les services dans un registre commun, et de simplifier les fonctionnalités de recherche et de publication des services Web. La découverte des services est assurée par un annuaire UDDI (Universal Description, Discovery, and Integration)[117]. La recherche et la sélection dans UDDI reposent sur la publication préalablement décrite du service et de son fournisseur. Par la suite, le client d'un service web connaitra les fournisseurs d'un service, les services proposés par un fournisseur donné, les moyens d'invoquer un service. Pour répondre aux besoins des clients, UDDI organise l'ensemble des informations qu'il contient en trois parties, spécifiées en XML. Chacune partie peut être utilisée pour faire une recherche via UDDI. Ces parties sont les suivantes[38] :

— Les pages blanches (White Paper). Ce composant permet de connaître les informations à propos de l'organisation proposant le service. Cette description contient toutes les informations jugées pertinentes pour identifier l'organisation (telles que son nom, son adresse physique). Le futur client du service retrouve dans les pages blanches les informations que le fournisseur a renseignées dans l'élément Business Entity lors de la publication.

— Les pages jaunes (Yellow Paper). Les pages jaunes de l'annuaire UDDI détaillent la description de l'organisation faite dans les pages blanches en répertoriant les services proposés. Dans cette section, sont décrits : la catégorie de l'entreprise, le secteur d'activité dans lequel exerce l'entreprise, les services offerts par cette organisation, le type de ser-

vices et les conventions d'utilisation – prix, qualité de service, etc. Cette description repose sur la classification standard de l'industrie nord-américaine (NAICS et UNSPSC). La description des services contenue dans les pages jaunes est non technique et est renseignée par les fournisseurs eux-mêmes.

— Les pages vertes (Green Paper). Les pages vertes comportent les informations techniques liées aux services Web et basées sur leur description WSDL.

2.6.5 Couche composition des services Web

Cette couche sert à traiter le processus de composition qui permet d'agréger des services Web. La composition permet de construire de nouveaux services appelés services composites par assemblage de services déjà existants. Les services participant à la composition sont nommés élémentaires ou atomiques. Une composition de service est un concept fondamental dans l'informatique orientée services dans lequel, un service composite est une agrégation de services collectivement composés pour automatiser une tâche ou un processus métier particulier. Nous distinguons deux concepts de composition de service, «orchestration» et «chorégraphie» des services Web. Les deux concepts impliquent la coordination ou le contrôle du fait que les services Web individuels fonctionnent ensemble pour former un processus global cohérent.

2.6.6 Couches Verticles (Transaction, Securité, ..)

Ces couches rendent viables l'utilisation effective des services web dans le monde le monde industriel. Le consortium W3C a proposé plusieurs spécifications [39] pour gérer les problèmes de sécurité, de transactions, de gestion de messages.

2.7 Composition des services web

La composition de services Web consiste à regrouper plusieurs services en un seul service afin d'exécuter des fonctions plus complexes [1].

Service atomique

Nous distinguons les services Web atomiques et composites. Un service atomique (également appelé service élémentaire[40] est un point d'accès à une application qui ne dépend pas d'un autre service Web pour satisfaire les demandes des utilisateurs.

Service composite

Un service composite est une structure parapluie qui rassemble d'autres services composites et atomiques qui collaborent pour mettre en œuvre un ensemble d'opérations [41][42] [43]. Les services regroupés par un service composite sont appelés services de composants. Un exemple de service composite serait un service de préparation de voyages intégrant des services de réservation de vols, de réservation d'hôtels, de recherche, etc.

Qu'il soit atomique ou composite, un service Web est spécifié par un URL, un ensemble d'attributs et un ensemble d'opérations. Les attributs d'un service fournissent des informations utiles pour les consommateurs finaux du service.

Orchestration et Chorégraphie

L'ensemble standard de technologies de service Web (XML, SOAP et WSDL) fournit les moyens de décrire, de localiser et d'appeler un service Web en tant qu'entité à part entière. Bien qu'un service Web puisse exposer de nombreuses opérations, chaque fichier WSDL décrit des fonctions de bas niveau assez atomiques. Il y a deux façons de décrire la séquence d'activités qui composent un processus métier : l'orchestration de service et la chorégraphie de service : L'orchestration de service représente un seul processus métier exécutable qui coordonne l'interaction entre les différents services, en décrivant un flux du point de vue et sous le contrôle d'un seul point de terminaison. La norme pour l'orchestration de services Web la plus supportée par l'industrie est BPEL.

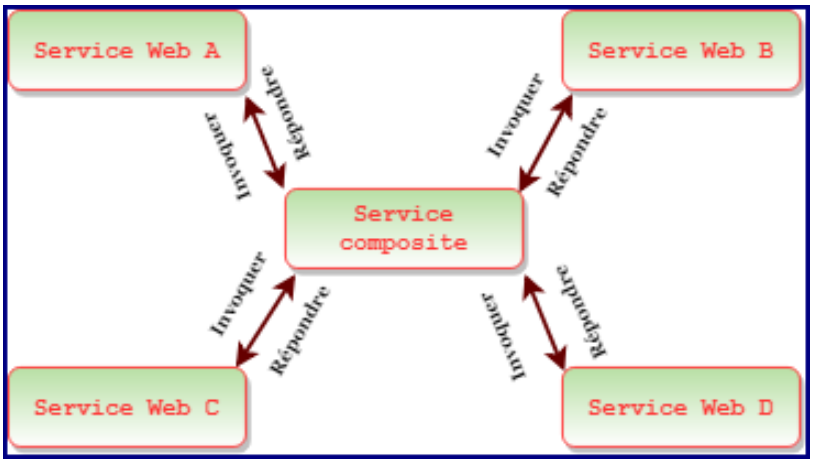

FIGURE 2.6 – Approche orchestration[1]

La chorégraphie est généralement associée aux interactions qui se produisent entre plusieurs services Web plutôt qu'à un processus métier spécifique qu'une seule partie exécute. Le mécanisme de chorégraphie est pris en charge par le standard WS-CDL (Web Services Choreography Description Language).

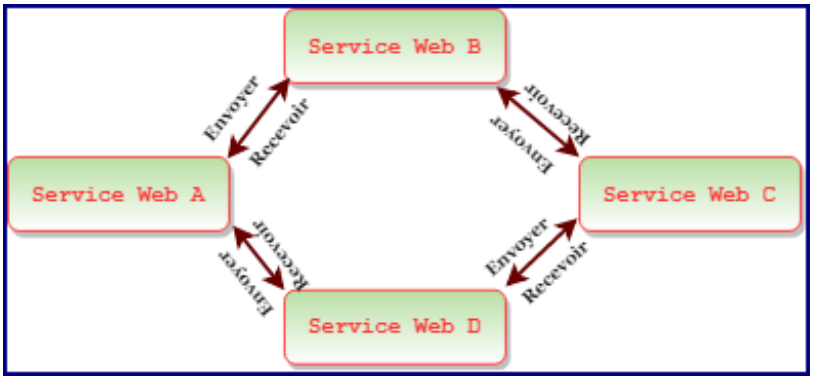

FIGURE 2.7 – Approche Chorégraphie[1]

L'orchestration de service représente toujours le contrôle du point de vue d'une partie. Cela diffère de la chorégraphie de service, qui est plus collaborative et permet à chaque partie impliquée de décrire sa part dans l'interaction. La chorégraphie représente une description globale du comportement observable de chacun des services participant à l'interaction, qui est définie par l'échange public de messages, de règles d'interaction et d'accords entre deux ou plusieurs points de terminaison de processus métier. La chorégraphie est particulièrement utile

dans les situations où plusieurs parties doivent collaborer, mais aucune d'elles ne veut prendre la responsabilité d'exécuter une orchestration centralisée[19]. Elle est généralement associée aux interactions qui se produisent entre plusieurs services Web plutôt qu'à un processus métier spécifique qu'une seule partie exécute.

Types de composition des services web

On distingue deux types de composition :

Composition statique qui consiste à générer manuellement les flux de composition. Dans une composition statique, l'agrégation des services a lieu au moment de la conception. Les composants de service requis pour la composition sont choisis, liés ensemble, puis déployés.

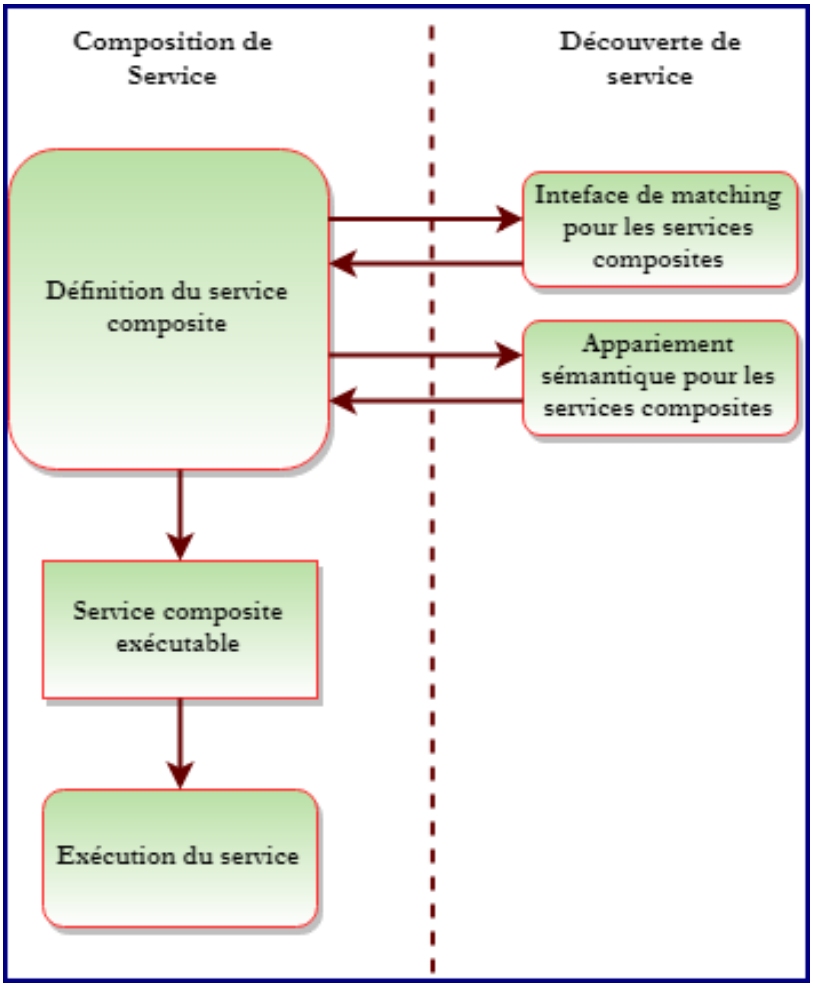

FIGURE 2.8 – Composition statique[44]

La composition statique n'est pas flexible et adaptable dans les cas où il y a des changements fréquents d'exigences ou des services qui ne peuvent pas être prévus au moment de la conception. Une fois qu'un service de liaison anticipée devient indisponible, ou s'il existe un meilleur service alternatif, la composition statique ne sera pas en mesure de fournir un meilleur support pour l'exécution du service composite en temps réel [1].

La composition dynamique qui s'articule sur des outils automatisant la génération de flux. Cette approche de composition de services Web nécessite que le système d'exécution prenne en charge la découverte, la sélection et la liaison automatiques des composants de service à partir des annuaires de services Web, tout en tenant compte des contraintes de l'utilisateur[45]

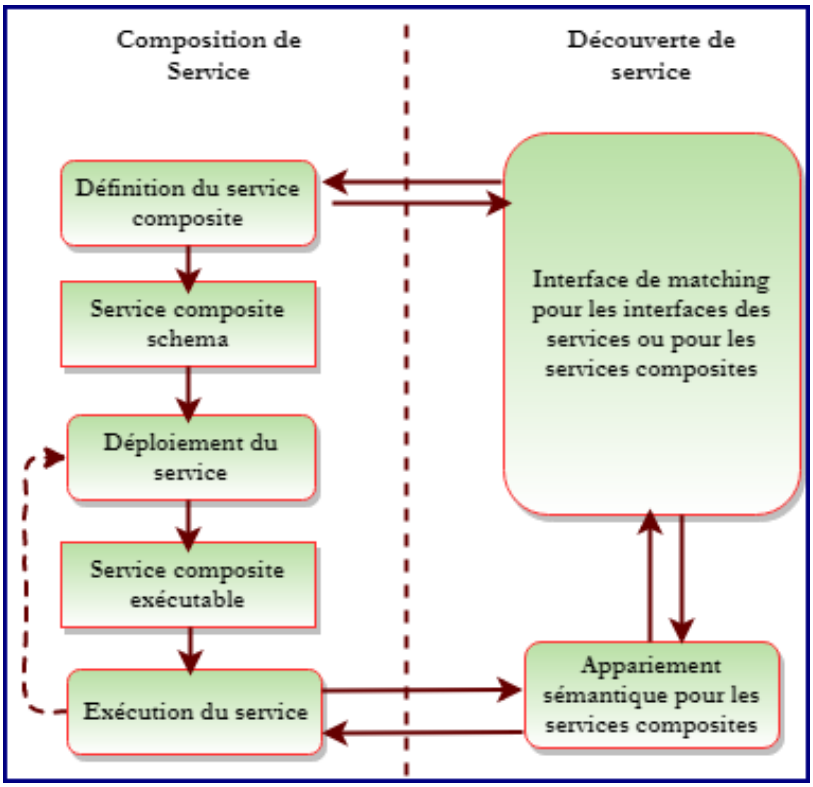

FIGURE 2.9 – Composition dynamique[44]

De cela, la composition dynamique est idéale car l'environnement de services Web est très dynamique par nature. Cependant, la composition dynamique des services est une tâche très difficile et un certain nombre de questions importantes doivent être prises en compte telles que la correction de la composition, les délais [1].

2.7.1 Composition manuelle, semi-automatisée et automatisée

La nature de la composition des services web peut être liée aussi au degré de participation de l'utilisateur dans le processus de composition à savoir : composition manuelle, semi-automatique ou automatique[45].

La composition manuelle suppose l'intervention totale de l'utilisateur sans l'intervention d'outils spécialisés. L'utilisateur se servira d'un simple éditeur texte pour programmer toutes les étapes du processus de la composition. Ce type de composition exige de l'utilisateur doit avoir le profil d'un développeur, faute de quoi, la composition devient une tâche fastidieuse où d'importants efforts sont à fournir de la part d'un utilisateur novice.

Par opposition à la composition manuelle, la composition automatique totalement automatisée prend en charge tout le processus de composition et le réalise automatiquement, sans qu'aucune intervention de l'utilisateur ne soit requise.

La composition semi-automatique permet à l'utilisateur de composer les services en s'appuyant sur des outils dédiés. Cette façon de faire permet à l'utilisateur de garder un certain contrôle et de superviser le processus de composition sans qu'il ait pour autant un prol de programmeur en s'appuyant sur l'assistance fournie par les outils graphiques.

2.7.2 Cycle de vie d'une composition

Le W3C a fourni un cycle de vie de service Web, mais les services Web dynamiques composés sont plus complexes que les services Web préexistants [46]. D'après[1], Un cycle de vie d'une composition de services Web reposant à partir de six activités :

— **Phase de définition :** Pendant cette phase, le demandeur de service spécifie l'exigence de composition des services, qui devrait fournir suffisamment d'informations sur les besoins et les préférences des utilisateurs pour le service composite. L'exigence est ensuite décomposée, semi-automatiquement ou automatiquement, en un modèle de processus abstrait (le service composite abstrait), qui spécifie un ensemble d'activités, le contrôle et le flux de données entre eux, les exigences de qualité de service et les comportements exceptionnels.

— **Phase de sélection du service :** Dans cette phase, pour chaque activité dans le service

composite, les services Web appropriés qui correspondent aux exigences de l'activité sont localisés en effectuant une recherche dans le registre de services, sur la base des informations contenues dans les documents de description de service publiés. Il est probable que plus d'un service candidat satisfera aux exigences. Par conséquent, le service le mieux adapté doit être sélectionné. Une fois que tous les services Web requis sont identifiés et liés aux activités correspondantes, le service composite construit est produit.

— **Phase de déploiement :** Dans cette phase, le service composite construit est déployé pour permettre son instanciation et son invocation par les utilisateurs finaux. Le résultat de cette phase est le service composite exécutable.

— **Phase d'exécution :** Dans cette phase, l'instance de service composite sera créée et exécutée par le moteur d'exécution, qui est également responsable de l'appel des composants de service individuels. Pendant l'exécution de l'instance de service composite, les tâches de surveillance, y compris la consignation, le suivi de l'exécution, la mesure de la performance et la gestion des exceptions, doivent être effectuées.

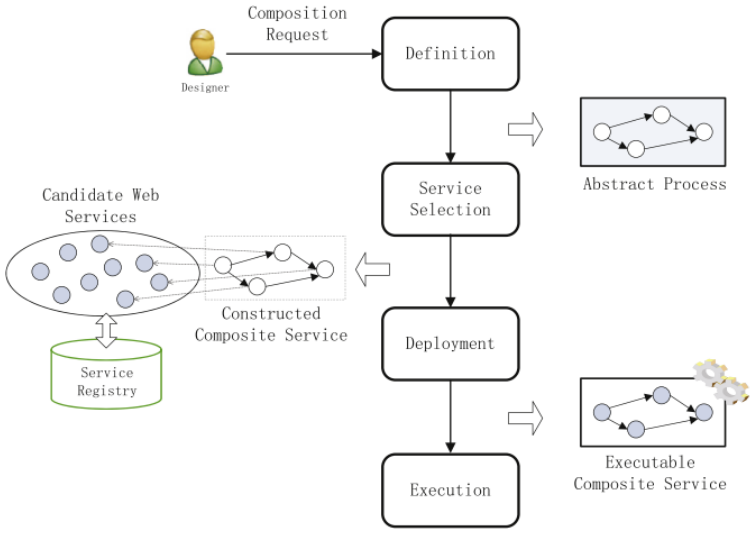

FIGURE 2.10 – Cycle de vie d'une composition[44]

2.7.3 Langages de composition des services Web

Plusieurs langages ont été proposés pour la composition des services Web. Le langage d'exécution des processus métier bénéficie du soutien le plus important des principaux fournisseurs,

notamment Microsoft et IBM. Dans ce qui suit, nous allons décrire quelques langages de composition :

BPEL4WS

BPEL4WS ou BPEL (Business Process Execution Language) est un langage basé sur XML pour la spécification et l'exécution de processus métier utilisant les services Web. Son développement a débuté en 2002 par IBM, Microsoft dont les concepts de la langue sont basés sur deux anciennes langues, WSFL (Web Services Flow Language) d'IBM et XLANG de Microsoft. WSFL a été conçu pour la composition de Web Services et est basé sur le concept de graphes orientés. Dans ce concept, les activités sont représentées par des nœuds et connectées par des liens indiquant le flux de contrôle. XLANG servait à l'origine de langage de modélisation pour les processus métier, il est basé sur le concept de la structuration en blocs comme familier de la plupart des langages de programmation impératifs actuels pour modéliser le flux de contrôle entre les activités. BPEL combine et étend les capacités de ses deux langages prédécesseurs [25].

WS-CDL (Web Services Choreography Description Language)

WS-CDL est un langage basé sur XML qui décrit les collaborations pair-à-pair des participants en définissant leur comportement observable. WS-CDL capture les interactions de service d'un point de vue global, ce qui signifie que tous les services participants sont traités de manière égale, ce qui est différent de BPEL où les interactions de service sont décrites du point de vue d'un seul participant. WS-CDL prend en charge la gestion des exceptions via un type d'unité de travail spécial, à savoir Exception, qui est associé à une chorégraphie et qui est activé lorsqu'une exception se produit[1].

BPML (Business Process Modeling Language)

BPML est un langage XML proposé par Business Process Management Initiative. Dans sa dernière version, il incorpore de nombreux concepts d'interface WSCI (Web Service Choreography Interface), qui se concentre sur la chorégraphie des services Web après qu'il a été conçu pour décrire les processus métier pouvant être exécutés par un système de gestion des processus métier, BPML fournit des activités basiques et structurelles similaires à celles de BPEL. BPML permet aux développeurs de planifier des tâches pour déterminer l'heure d'exécution de la tâche en utilisant la planification. Le contexte est un élément important dans BPML[1].

BPML fournit également le mécanisme de gestion des exceptions (processus d'exception)

pour gérer les événements exceptionnels et le mécanisme de compensation (processus de compensation) pour inverser les effets d'une activité terminée.

2.8 Conclusion

Nous avons présenté dans ce chapitre, une vue globale sur les services web tout en relatant leurs technologies et leur fonctionnement. En d'autres termes, les services Web ne sont pas concernés par la façon dont les messages sont produits ou consommés par des programmes.

Nous avons détailleé l'architecture des services web en décrivant toutes les couches qui la composent. Nous avons aussi détaillé la couche composition des services Web qui permet de combiner des services existants pour former de nouveaux services plus élaborés.

Nous avons clôturé le chapitre en abordant l'aspect non fonctionnel caractérisé par les paramètres de qualité de services (QoS) qui jouent un rôle primordial dans le processus de découverte. Cette dernière est très attachée à la phase de sélection qui fera l'objet du chapitre suivant.

Chapitre 3

Logique Floue

3.1 Introduction

La logique classique joue un rôle immense dans différents domaines, sa structure peut être exprimer des faits qu'avec "vrai" ou "faux" qui limite son champ d'action dans des techniques et des applications comme l'Intelligence Articielle, la prise de décision des données incertaines, etc. qui veulent imiter le raisonnement et l'esprit humain. C'est à dire des techniques qui s'appuient sur l'incertitude pour leur bon fonctionnement. La logique oue est justement conçue pour régler ce problème, pour permettre la caractérisation des éléments de façon "graduelle". Notre perception du monde réel est imprégnée de concepts qui n'ont pas de limites clairement définies, par exemple, beaucoup, grands, beaucoup plus grands que les jeunes, etc. ne sont vrais que dans une certaine mesure et ils sont faux à un certain degré. Ces concepts peuvent être appelés concepts flous ou vagues, un cerveau humain travaille avec eux, tandis que les ordinateurs ne peuvent pas le faire. Les langages naturels, dont le niveau est beaucoup plus élevé que les langages de programmation, sont flous alors que les langages de programmation ne le sont pas[47]. L'utilité des ensembles flous réside dans leur capacité à modéliser des données incertaines ou ambiguës, si souvent rencontrées dans la vie réelle. L'idée de logique floue est similaire au processus de perception et d'inférence de l'être humain.

Le terme de logique floue a été introduit avec la proposition de 1965 de la théorie des ensembles flous par Lotfi Zadeh en 1965 à l'université de Berkeley comme extension de la logique booléenne en se basant sur sa théorie mathématique des ensembles flous, qui est une généra-

lisation de la théorie des ensembles classiques. En introduisant la notion de degré dans la vérification d'une condition, permettant ainsi à une condition d'être dans un autre état que vrai ou faux, la logique floue confère une flexibilité très appréciable aux raisonnements qui l'utilisent, ce qui rend possible la prise en compte des imprécisions et des incertitudes[48].

La logique floue n'est pas une logique imprécise, mais une logique qui s'adapte à l'humain en laissant une place entre la certitude de la vérité et la certitude du faux. Il ressemble au raisonnement humain dans l'utilisation d'informations approximatives et incertaines pour prendre des décisions[50].

Cette logique ne consiste pas à être précis dans les affirmations, mais au contraire répondre à des propositions vagues, nécessitant une certaine incertitude. Par exemple, en logique classique, à la question : Est-ce que cette personne est grande? On ne peut répondre que par vrai, si c'est le cas, ou faux dans le cas contraire. Avec la logique floue, on peut représenter les cas où la personne est très petite, moyennement petite, normale, pas très grande, grande,etc[51].

Un des intérêts de la logique floue pour formaliser le raisonnement humain est que les règles sont énoncées en langage naturel. Elle est devenue un outil important pour le nombre d'applications différentes allant du contrôle du système d'ingénierie à l'intelligence artificielle[52].

3.2 Historique de la logique floue

La logique floue a été marque par plusieurs dates[118] :

— En 1965, Lotfi A. Zadeh, de l'Université de Californie à Berkeley, publie «Fuzzy Sets», qui expose les mathématiques de la théorie des ensembles flous et, par extension, de la logique floue. Zadeh avait observé que la logique informatique conventionnelle ne pouvait pas manipuler des données qui représentaient des idées subjectives ou vagues, donc il a créé une logique floue pour permettre aux ordinateurs de déterminer les distinctions entre les données avec des nuances de gris, semblables au processus du raisonnement humain.

— Seiji Yasunobu et Soji Miyamoto d'Hitachi ont suscité l'intérêt pour les systèmes flous. En 1985, ils ont fourni des simulations qui démontraient la supériorité des systèmes de contrôle flou pour le chemin de fer de Sendai. Leurs idées ont été adoptées, et les sys-

tèmes flous ont été utilisés pour contrôler l'accélération et le freinage lorsque la ligne a ouvert en 1987.

— Toujours en 1987, lors d'une réunion internationale de chercheurs flous à Tokyo, Takeshi Yamakawa a démontré l'utilisation du contrôle flou, à travers un ensemble de puces logiques floues dédiées, dans une expérience de "pendule inversé". C'est un problème de contrôle classique, dans lequel un véhicule essaie de maintenir un poteau monté sur son sommet par une charnière en se déplaçant d'avant en arrière.

— Les observateurs ont été impressionnés par cette démonstration, ainsi que par des expériences ultérieures de Yamakawa au cours desquelles il a monté un verre de vin contenant de l'eau ou même une souris vivante au sommet du pendule.

— Le système a maintenu la stabilité dans les deux cas. Yamakawa a finalement organisé son propre laboratoire de recherche sur les systèmes flous pour l'aider à exploiter ses brevets sur le terrain.

3.3 Y a-t-il un besoin de logique floue

La logique floue a un niveau de généralité beaucoup plus élevé que la logique bivalente. C'est la généralité de la logique floue qui sous-tend une grande partie de ce que la logique floue a à offrir. Parmi les contributions importantes de la logique floue sont les suivants[53]

— **FL-généralisation.** Toute théorie basée sur la logique bivalente, T, peut être généralisée FL, et donc améliorée, en ajoutant à T des concepts et des techniques tirés de la logique floue. Exemples : contrôle flou, programmation linéaire floue, théorie des probabilités floues et topologie floue.

— **Variables linguistiques et règles if-then floues.** Le formalisme des variables linguistiques et des règles if-then floues est, en effet, un puissant langage de modélisation largement utilisé dans les applications de la logique floue. Fondamentalement, le formalisme sert de moyen de synthèse et de compression de l'information par l'utilisation de la granulation.

— **Précision co-intensive.** La logique floue a un grand pouvoir de précision. Ce pouvoir est nécessaire pour la formulation de définitions communes de concepts scientifiques

et la formalisation conjointe de domaines centrés sur l'homme tels que l'économie, la linguistique, le droit, la résolution de conflits, la psychologie et la médecine.

— **NL-Calcul (computing with words).** La logique floue sert de base au calcul NL, c'est-à-dire au calcul avec des informations décrites en langage naturel. NL-Calcul est directement lié à la mécanisation de la compréhension du langage naturel et au calcul avec des probabilités imprécises. Plus généralement, le calcul NL est nécessaire pour traiter l'incertitude de second ordre, c'est-à-dire l'incertitude sur l'incertitude ou l'incertitude pour le court terme.

3.4 Définition de la logique Floue

Plusieurs défintions ont été abordées pour le mot logique floue. dans la suite, nous allons donner quelques définitions :

Définition de Zadeh :

La logique floue s'intéresse aux principes formels du raisonnement approximatif, avec un raisonnement précis considéré comme limitant. il vise à modéliser le mode de raisonnement imprécis qui joue un rôle essentiel dans la capacité humaine remarquable à prendre des décisions dans un environnement d'incertitudes et d'imprécisions [54].

Définition de Wikipédia :

La logique floue est une forme de logique de grande valeur dans laquelle les valeurs de vérité des variables peuvent être n'importe quel nombre réel entre 0 et 1. En revanche, dans la logique booléenne, les valeurs de vérité des variables ne peuvent être que la valeur entière 0 ou 1. La logique floue a été utilisée pour gérer le concept de vérité partielle, où la valeur de la vérité peut varier entre complètement vrai et complètement faux[119]. En outre, lorsque des variables linguistiques sont utilisées, ces degrés peuvent être gérés par des fonctions spécifiques (appartenance).

une autre définition :

La logique floue est une méthodologie de résolution de problèmes qui fournit un moyen simple de tirer des conclusions définitives à partir d'informations vagues et imprécises[52].

3.5 Domaines d'application :

La majorité des premières applications réussies de la logique floue ont été mises en œuvre au Japon. La première application notable était sur le train à grande vitesse à Sendai, dans lequel la logique floue pouvait améliorer l'économie, le confort et la précision du trajet. Il a également été utilisé en reconnaissance des symboles écrits à la main dans les ordinateurs de poche de Sony, L'aide au vol pour les hélicoptères, le contrôle des systèmes de métro afin d'améliorer le confort de conduite, la précision de l'arrêt et l'économie d'énergie, la consommation de carburant améliorée pour les automobiles, le contrôle à un bouton pour les machines à laver, le contrôle automatique du moteur pour les aspirateurs avec reconnaissance de l'état de la surface et Le degré de souillure et les systèmes de prédiction pour la reconnaissance précoce des tremblements de terre à travers l'Institut de sismologie Bureau de météorologie au Japon.

La logique floue a potentiellement de nombreuses applications dans tous les domaines, les principaux domaines d'applications de logique floue sont [56] :

- Contrôle automatique des vannes de barrage pour centrales hydroélectriques (Tokyo Electric Power.)
- Contrôle simplifié des robots (Hirota, Fuji Electric, Toshiba, Omron)
- Caméra visant la télédiffusion d'événements sportifs (Omron)
- Contrôle efficace et stable des moteurs de voitures (Nissan)
- Régulateur de vitesse pour automobiles (Nissan, Subaru)
- Substitution d'un expert pour l'évaluation des activités boursières (Yamaichi, Hitachi)
- Planification optimisée des horaires d'autobus (Toshiba, NipponSystem, Keihan-Express)
- Système d'archivage pour documents (Mitsubishi Elec.)
- Système de prévision pour la reconnaissance précoce des tremblements de terre (Bureau de sismologie de métrologie, Japon)
- Technologie de la médecine : diagnostic du cancer (École de médecine de Kawasaki)
- Reconnaissance des motifs en images avec des caméras vidéo (Canon, Minolta)
- Commande automatique des moteurs pour aspirateurs avec reconnaissance de l'état de surface et du degré de salissure (Matsushita) Commande de rétro-éclairage pour caméscopes (Sanyo)

3.6 Ensembles flous

Les ensembles flous fournissent des moyens de modéliser l'incertitude associée au flou, à l'imprécision et au manque d'information concernant un problème ou une usine, etc.

Exemple : Considérons la signification d'une «personne de grande taille». Pour une personne X, la personne de grande taille peut être une personne dont la taille est supérieure à 1m80 ". Pour l'autre individu Y, la personne de grande taille peut être une personne dont la taille est supérieure ou égale à 1m70 ". Ce mot "grand" est appelé comme un descripteur linguistique. Le terme «grand» donne le même sens aux individus X et Y, mais on constate qu'ils ne fournissent pas tous deux une définition unique[47].

Les idées floues et la logique floue sont si souvent utilisées dans notre vie quotidienne que personne n'y prête attention.

Definition d'un ensemble flou :

Un ensemble flou $\widetilde{A}$ est un sous-ensemble d'un univers de discours X, caractérisé par une fonction d'appartenance $u_{\widetilde{A}}(x)$ représentant un mappage $u_{\widetilde{A}} \rightarrow [0,1]$. La valeur de fonction $u_{\widetilde{A}}(x)$ est appelée la valeur d'appartenance, qui représente le degré de vérité que x est un élément de l'ensemble flou. On suppose que, $u_{\widetilde{A}}(x) \in [0,1]$ où $u_{\widetilde{A}}(x) = 0$ révèle que x appartient complètement à, tout en indiquant que x n'appartient pas à l'ensemble flou $\widetilde{A}$[57].

Les ensembles classiques sont des ensembles sans ambiguïté dans leur appartenance. La théorie des ensembles flous est une théorie très efficace pour traiter les concepts d'ambiguïté.

Dans l'ensemble classique, sa fonction caractéristique assigne une valeur de 1 ou 0 à chaque individu dans l'ensemble universel, là en discriminant entre les membres et les non-membres de l'ensemble numérique considéré. Les valeurs assignées aux éléments de l'ensemble universel se situent dans une plage spécifiée et indiquent la qualité d'appartenance de ces éléments dans l'ensemble. Les plus grandes valeurs indiquent des degrés plus élevés d'appartenance à un ensemble, une telle fonction est appelée une fonction d'appartenance et l'ensemble est défini par un ensemble flou.

Un ensemble flou est donc un ensemble contenant des éléments ayant différents degrés d'appartenance à l'ensemble. Cette idée contraste avec le classique ou crisp, car les membres

d'un ensemble réel ne seraient pas membres à moins que leur adhésion soit complète dans cet ensemble (appartenance a une valeur de 1). Les éléments d'un ensemble flou, parce que leur appartenance n'a pas besoin d'être complète, peuvent également être membres d'un autre ensemble flou sur le même univers. L'ensemble flou est mappé à une valeur numérique réelle dans l'intervalle 0 à 1. Si un élément de l'univers, disons x, est un membre de l'ensemble flou A, le mappage est donné par $u(x) \in [0,1]$.

Comparé à un ensemble classique, un ensemble flou permet aux membres d'avoir une frontière. En d'autres termes, un ensemble flou permet à un membre d'appartenir à un ensemble à un degré partiel.

L'ensemble flou est un outil puissant qui nous permet de représenter des objets ou des membres d'une manière vague ou ambiguë. L'ensemble flou fournit également une manière qui est similaire aux concepts et au processus de pensée d'un être humain. Cependant, seul l'ensemble flou lui-même ne peut mener à aucun produit utile et pratique tant que le processus d'inférence floue n'est pas appliqué[58].

3.7 Variable linguistique

Une variable linguistique est une variable ayant une valeur qui est une expression de langage naturel faisant référence à une certaine quantité d'intérêt. Les variables linguistiques sont utilisées dans les activités quotidiennes ordinaires.

Ces expressions de langage naturel sont ensuite à leur tour des noms pour des ensembles flous composés des valeurs numériques possibles que la quantité d'intérêt peut assumer[56]. Un mot ou une phrase de mot est la principale différence entre une variable linguistique et une variable numérique.

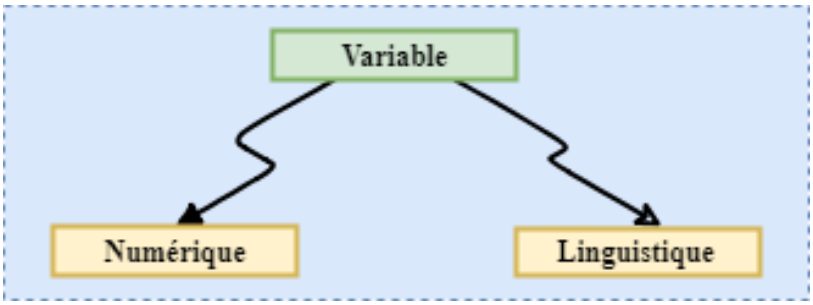

FIGURE 3.1 – variable linguistique

Les variables linguistiques associent une condition linguistique à une variable numérique.

Une variable numérique est le type de variable qui est utilisé dans la plupart des programmes informatiques. Par contre, une variable linguistique a un caractère proportionnel : dans toutes les implémentations logicielles de variables linguistiques, elles sont représentées par des valeurs fractionnaires de l'ordre de 0 à 1[59].

Selon [60],une variable linguistique est un terme composite $u = u_1, u_2, \ldots, u_n$ qui est une concaténation de termes atomiques $u_1, u_2, \ldots, u_n$. Ces termes atomiques peuvent être divisés en quatre catégories :

— les termes principaux, qui sont les étiquettes des sous-ensembles flous spécifiés de l'univers du discours (par exemple petit et grand) ;

— Les connecteurs AND, OR et la négation NOT ;

— des haies telles que **Très**, **plus**, **Plutôt**, **légèrement**,**Plus ou mois**, etc . ;

— des marqueurs tels que des parenthèses.

3.8 Opérations des ensembles flous

Le concept de sous-ensemble flou est une généralisation du concept d'ensemble classique, on est conduit à introduire des opérations sur les sous-ensembles flous qui sont équivalentes aux opérations classiques de la théorie des ensembles lorsqu'on a affaire à des fonctions d'appartenance à valeurs 0 ou 1. Dans ce qui suit, nous allons présenter les opérations les plus couramment utilisées[62].

Inclusion

Un ensemble flou $\widetilde{A} \subset U$ est inclu dans un autre sous ensemble $\widetilde{B} \subset U$ si et seulement si tout élément x de U qui appartient à $\widetilde{A}$ appartient aussi à $\widetilde{B}$ avec un degré au moins aussi grand.

Union

L'union de deux sous ensembles flous $\widetilde{A}$ et $\widetilde{B}$ de U est un ensemble flou constitué des éléments de U affectés du plus grand de leurs deux degrés d'appartenance, donné par $u_{\widetilde{A}}$ et $u_{\widetilde{B}}$.

C'est l'ensemble $u_{\widetilde{C}} = u_{\widetilde{A}} \cup u_{\widetilde{B}}$ de u tel que :

$\forall x \in U, u_{\widetilde{C}(x)} = max\{u_{\widetilde{A}(x)}, u_{\widetilde{B}(x)}\}$

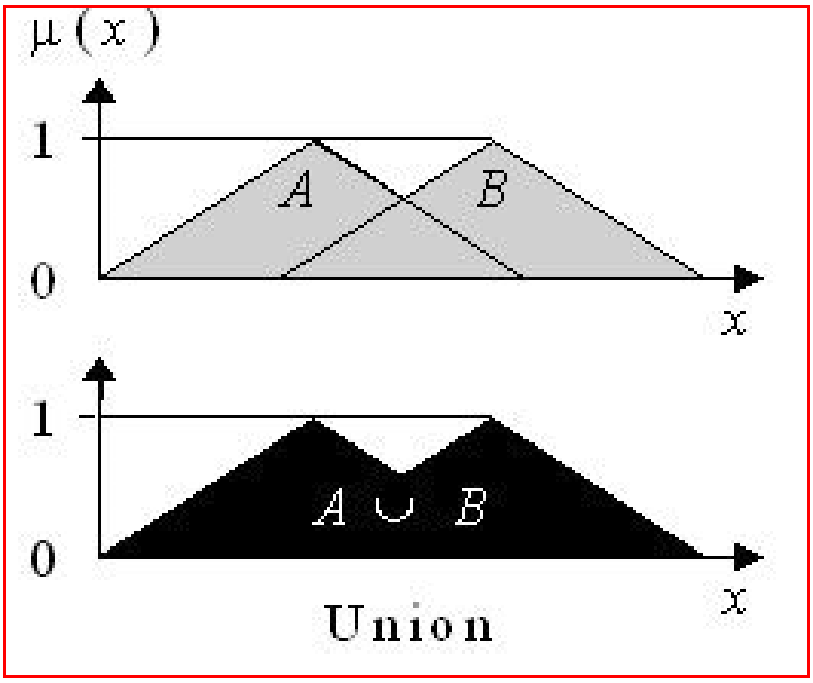

FIGURE 3.2 – Union de deux variables linguistiques

Intersection

L'intersection de deux sous ensembles flous $\widetilde{A}$ et $\widetilde{B}$ de U est un ensemble flou constitué des éléments de U affectés du plus petit de leurs deux degrés d'appartenance, donné par $u_{\widetilde{A}}$ et $u_{\widetilde{B}}$.

C'est l'ensemble $u_{\widetilde{C}} = u_{\widetilde{A}} \cap u_{\widetilde{B}}$ de u tel que :

$\forall x \in U, u_{\widetilde{C}(x)} = min\{u_{\widetilde{A}(x)}, u_{\widetilde{B}(x)}\}$

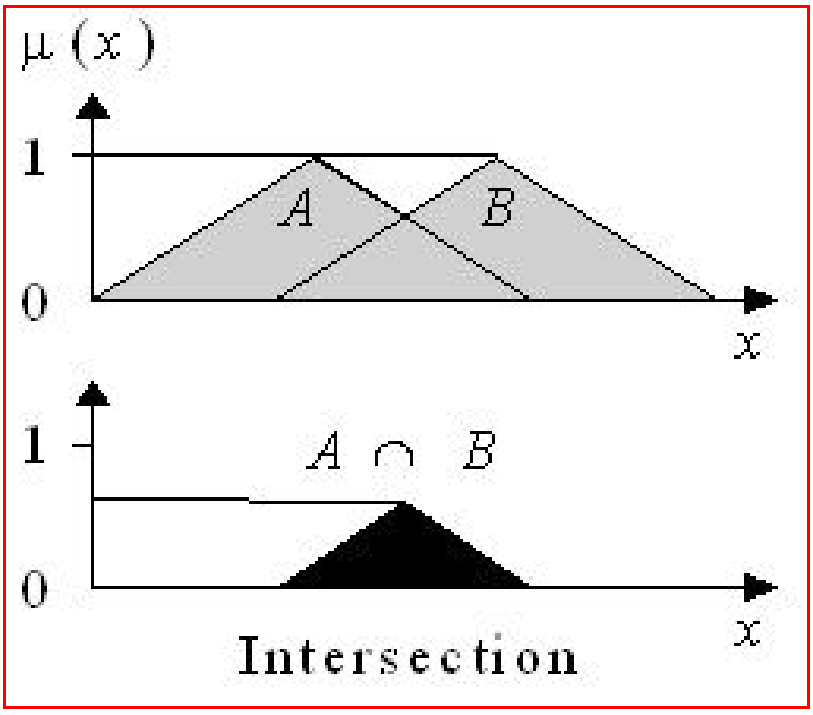

FIGURE 3.3 – intersection de deux variables linguistiques

Complément

Le complémentaire d'un sous-ensemble flou A de X noté $\bar{A}$ est défini par :

$u_{\widetilde{A}}(x) = 1 - u_{\widetilde{A}}(x)$

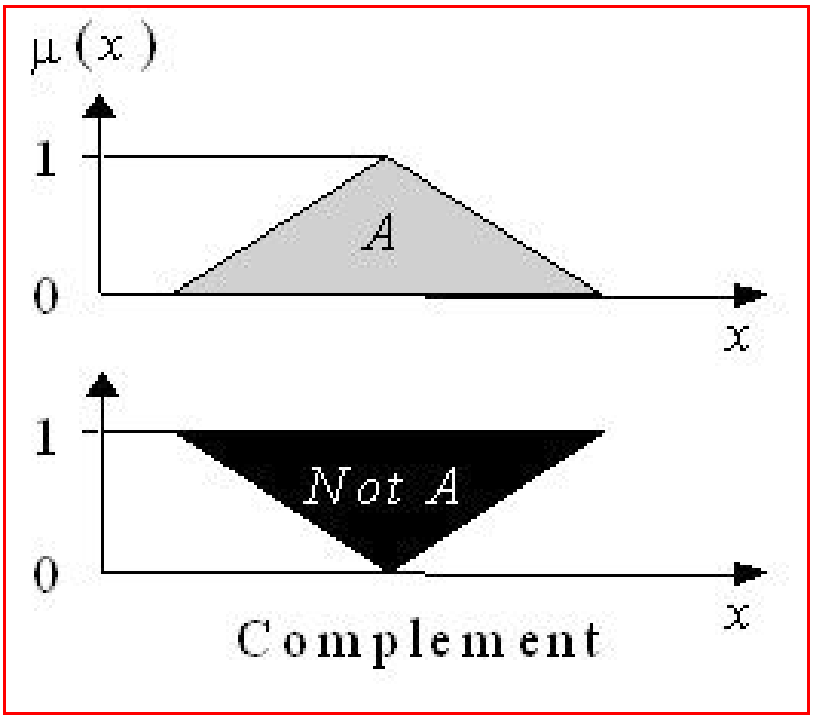

FIGURE 3.4 – Complément d'un ensemble flou

3.9 Propriétés des ensembles flous

Les propriétés de l'ensemble classique conviennent également pour les propriétés des ensembles flous. Les propriétés importantes de l'ensemble flou comprennent[47] :

Commutativité

$\widetilde{A} \cup \widetilde{B} = \widetilde{B} \cup \widetilde{A}$

$\widetilde{A} \cap \widetilde{B} = \widetilde{B} \cap \widetilde{A}$

Associativité

$\widetilde{A} \cup (\widetilde{B} \cup \widetilde{C}) = (\widetilde{A} \cup \widetilde{B}) \cup \widetilde{C}$

$\widetilde{A} \cap (\widetilde{B} \cap \widetilde{C}) = (\widetilde{A} \cap \widetilde{B}) \cap \widetilde{C}$

Distributivité

$\widetilde{A} \cup (\widetilde{B} \cap \widetilde{C}) = (\widetilde{A} \cup \widetilde{B}) \cap (\widetilde{A} \cup \widetilde{C})$

$\widetilde{A} \cap (\widetilde{B} \cup \widetilde{C}) = (\widetilde{A} \cap \widetilde{B}) \cup (\widetilde{A} \cap \widetilde{C})$

Idempotence

$\widetilde{A} \cup \widetilde{A} = \widetilde{A}$

$\widetilde{A} \cap \widetilde{A} = \widetilde{A}$

Identité

$\widetilde{A} \cup \emptyset = \widetilde{A}$

$\widetilde{A} \cap X = \widetilde{A}$

$\widetilde{A} \cap \emptyset = \emptyset$

$\widetilde{A} \cup X = X$

Transitivité

Si $\widetilde{A} \subset \widetilde{B} \subset \widetilde{C}$ Alors $\widetilde{A} \subset \widetilde{C}$

involution

$\overline{\overline{\widetilde{A}}} = \widetilde{A}$

3.10 Fonction d'appartenance

L'ensemble classique (net) est défini de manière à dichotomiser les individus dans un univers donné de discours en deux groupes : les membres et les non-membres.

Un ensemble flou peut être défini mathématiquement en assignant à chaque individu possible dans l'univers du discours une valeur représentant son degré d'appartenance à l'ensemble flou. Un sous-ensemble flou est complètement défini par la donnée de sa fonction d'appartenance.

La logique floue admet des degrés d'appartenance à un ensemble donné. Le degré d'appartenance à un ensemble flou est matérialisé par un nombre compris entre 0 et 1. Une valeur précise de la fonction d'appartenance liée à une valeur de la variable est notée u et appelée facteur d'appartenance [61].

Caractéristiques d'une fonction d'appartenance

Une fonction d'appartenance a plusieurs caractéristiques[55][62] :

1. **Le type :** la forme du nombre ou qui peut être triangulaire, trapézoïdale, gaussienne ou sigmoïdale.

2. **Support :** Le support d'un sous-ensemble flou de A de X , noté $Supp(A)$, est l'ensemble de tous les éléments qui lui appartiennent au moins un petit peu.
3. **Hauteur :** La hauteur du sous-ensemble flou A de X , notée $h(A)$, est le plus fort degré avec lequel un élément de X appartient à A.
4. **Noyau :** Le noyau d'un sous ensemble flou A de X , noté $Noy(A)$, est l'ensemble de tous les éléments qui lui appartiennent totalement (avec un degré 1).
5. **Cardinalité :** La cardinalité d'un sous-ensemble flou A de X, noté A, est le nombre d'éléments appartenant à A pondéré par leur degré d'appartenance.

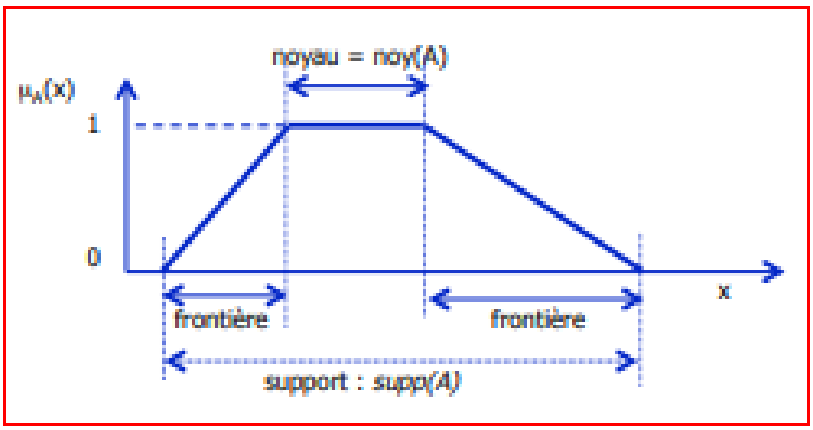

FIGURE 3.5 – caractéristiques d'une fonction d'appartenance[63]

Types de fonctions d'appartenance

En pratique, les fonctions d'appartenance peuvent avoir plusieurs types différents, tels que la forme d'onde triangulaire, la forme d'onde trapézoïdale, la forme d'onde gaussienne, la forme d'onde en cloche, la forme d'onde sigmoïdale..etc. Le type exact dépend des applications réelles.

Pour les systèmes qui nécessitent une variation dynamique significative sur une courte période de temps, une forme d'onde triangulaire ou trapézoïdale doit être utilisée [58].

Pour les systèmes nécessitant une précision de contrôle très élevée, une forme d'onde gaussienne ou en courbe S doit être sélectionnée.

Il existe plusieurs types des fonctions d'appartenance, les plus utilisés sont :

1. Fonction d'appartenance triangulaire

Un nombre triangulaire flou $\widetilde{N}$ est défini par un triplet (l, m, u) et la fonction d'appartenance $U_(\widetilde{N}(x)$ est définie par is defined as[64] :

$$\widetilde{N} = \begin{cases} 0 & \text{si } x < a \\ \frac{x-a}{b-a} & \text{si } a \leq x \leq b \\ \frac{c-x}{c-b} & \text{si } b \leq x \leq c \\ 0 & \text{si } x > c. \end{cases}$$

ou a, b, c sont des nombres réels et $(a < b < c)$

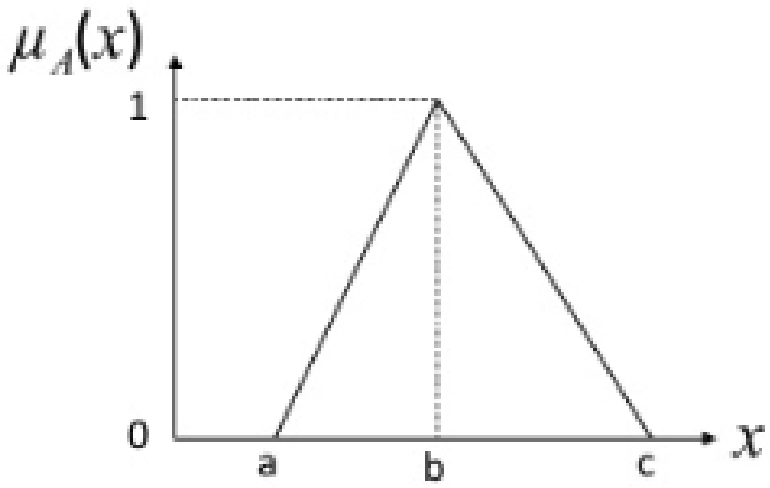

FIGURE 3.6 – Fonction d'appartenance triangulaire[57]

2. Fonction d'appartenance trapézoïdale

   Une fonction d'appartenance trapézoïdale, est dénie par quadruplet (a, b, c, d). Les paramètres a et b représentent respectivement la limite inférieure et la limite supérieure du support. Les paramètres c et d sont respectivement la borne inférieure et la borne

supérieure du noyau. Cette fonction est dénie par l'expression suivante :

$$\widetilde{N} = \begin{cases} \frac{x-b}{b-a} & \text{si } a < x \leq b \\ 1 & \text{si } b \leq x \leq c \\ \frac{d-x}{d-b} & \text{si } c \leq x \leq d \\ 0 & \text{autrement .} \end{cases}$$

Où a, b, c et d sont des nombres réels et $(a \leq b \leq c \leq d)$

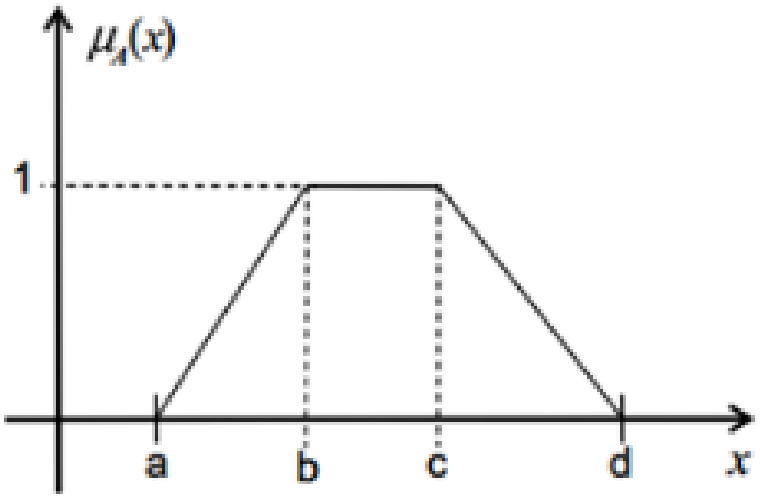

FIGURE 3.7 – Fonction d'appartenance trapezoidale[65]

3. Fonction d'appartenance gaussienne

   Une fonction d'appartenance gaussienne est caractérisée par sa valeur centrale m et son écart type σ.

   La fonction d'appartenance gaussienne est dénie par : $u_A(x) = exp^{(-\frac{1}{2}(\frac{x-m}{\sigma})^2)}$

   sa représentation est comme suit :

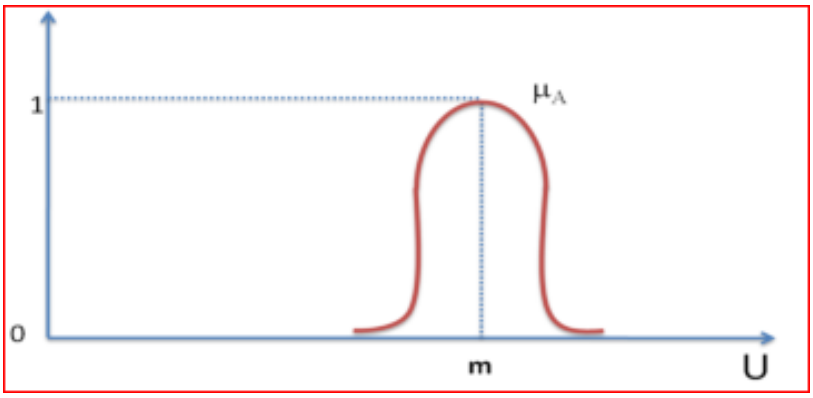

FIGURE 3.8 – Fonction d'appartenance gaussienne[63]

3.11 Système d'inférence flou

Le principe d'un système flou, c'est de pouvoir calculer des paramètres de sorties en fournissant au système un ensemble de règles formulés en langage naturel[120]. Pour implémenter un système d'inférence flou, trois étapes sont nécessaires[58] :

1. **Fuzzification :** Convertir des données classiques ou des données numériques en données floues ou fonctions d'appartenance (MF [1]) ;
2. **Processus d'inférence floue :** Combiner les fonctions d'appartenance avec les règles de contrôle pour dériver la sortie floue ;
3. **Defuzzification :** Utiliser différentes méthodes pour calculer chaque sortie associée et les mettre dans une table : la table de recherche. Décrochez la sortie de la table de recherche en fonction de l'entrée en cours pendant une application.

Ces trois étapes constituent ce qu'on appelle moteur d'inférence flou :

1. Memebership Function

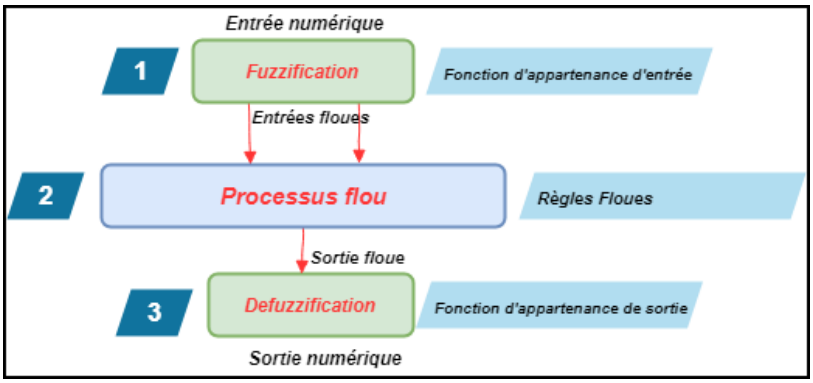

FIGURE 3.9 – Moteur d'inférence flou[56]

Fuzzification

Afin de permettre aux machines de gérer une entrée de langage vague, les entrées et sorties numériques doivent être converties en variables linguistiques avec des composants flous. Par exemple, pour qu'un client évalue les critères d'un service web en valeurs linguistiques, Ces critères doivent être convertis en variables linguistiques associées telles que **ÉLEVÉ**, **MOYEN**, **FAIBLE** et **RAPIDE**, ou **LENT**. Aussi, la valeur finale (de sortie) pour évaluer un tel service doit être convertie en floue, l'entrée et la sortie doivent également être converties à partir de données nettes en données floues. Cette étape de conversion est appelée fuzzification.

L'étape de fuzzification a pour but de transformer une donnée numérique en variable linguistique. Pour cela, le concepteur du système flou doit créer des fonctions d'appartenances qui permettent de définir le degré d'appartenance d'une donnée numérique à une variable linguistique.

Processus d'inférence floue

Une fois les variables numériques sont transformées en variables linguistiques, on va pouvoir les passer dans le moteur d'inférence. Ici, chaque règle du moteur d'inférence est écrite par le concepteur du système flou en fonction de connaissance qu'il possède. (Généralement, ces règles doivent d'être dictées par des experts pour donner plus de performance au système).

La règle de contrôle floue peut être considérée comme la connaissance d'un expert dans tout domaine d'application connexe. La règle floue est représentée par une séquence de la forme IF

THEN, conduisant à des algorithmes décrivant quelle action ou sortie doit être prise en termes d'informations actuellement observées, ce qui inclut l'entrée et la rétroaction si un système de contrôle en boucle fermée est appliqué[58].

La loi pour concevoir ou construire un ensemble de règles floues est basée sur la connaissance ou l'expérience d'un être humain, qui dépend de chaque application réelle différente.

Pour commencer le processus d'inférence floue, il est nécessaire de combiner les fonctions d'appartenance avec les règles de contrôle pour dériver la sortie de contrôle et d'organiser ces sorties. La règle de contrôle est le noyau du processus d'inférence floue, et ces règles sont directement liées à l'intuition et au sentiment d'un être humain.

Deffuzification

La défuzzification est le processus de conversion d'une sortie fuzzifiée en une seule valeur crisp par rapport à un ensemble flou. La valeur défuzzifiée dans un système d'inférence flou représente l'action à entreprendre pour contrôler le processus.

Lors de la seconde étape, on a généré un tas de commandes sous la forme de variables linguistiques (une commande par règle).

Le but de la défuzzification est de fusionner ces commandes et de transformer les paramètres résultants en donnée numérique.

L'étape de défuzzification se déroule en deux temps [120] : Dans la première étape, il faut fusionner les variables linguistiques communes à l'aide d'un opérateur de la logique floue choisi par le concepteur du système. Si on a plusieurs règles d'inférences qui génèrent plusieurs valeurs de la même variable linguistique, on peut choisir un opérateur pour combiner les valeurs de la variable. Cet opérateur sera dans la grande majorité des cas, le OU logique utilisant l'opérateur de maximalité.

Dans une deuxième étape, nous pouvons réellement entamer la partie délicate de la défuzzification. On a une série de variables linguistiques qui caractérisent une seule et même donnée. Ces variables linguistiques possèdent chacune une fonction d'appartenance. Défuzzifier une donnée revient donc à trouver la meilleure valeur quantitative en fonction des fonctions d'appartenances des variables linguistiques. pour faire cette étape de defuzzification, on aura besoin d'appliquer une technique de defuzzification.

3.12 Méthodes de defuzzification

Dans la littérature, il existe plusieurs méthodes de de défuzzification, par la suite, nous allons présenté seules qui sont couramment utilisées[56] :

Méthode Centre de gravité (COG)

Cette méthode fournit une valeur numérique basée sur le centre de gravité de l'ensemble flou. La zone totale de la distribution des fonctions d'appartenance utilisée pour représenter l'action de contrôle combinée est divisée en un certain nombre de sous-zones. La zone et le centre de gravité ou centroïde de chaque sous-zone sont calculés, puis la somme de toutes ces sous-zones est prise pour trouver la valeur défuzzifiée pour un ensemble flou discret.

Méthode Moyenne de maximum(MOM)

Dans cette méthode, la valeur défuzzifiée est considérée comme l'élément avec les valeurs d'appartenance les plus élevées. Lorsque plusieurs éléments ont des valeurs d'appartenance maximum, la valeur moyenne des maxima est prise.

Méthode moyenne pondérée (WAM)

Cette méthode est valable pour les ensembles flous avec des fonctions d'appartenance de sortie symétriques et produit des résultats très proches de la méthode COG ou COA. Cette méthode nécessite moins de calculs. Chaque fonction d'appartenance est pondérée par sa valeur d'adhésion maximale.

3.13 Conclusion

En résumé, un processus flou est un processus de crisp-fuzzy-crisp pour un système réel. L'entrée d'origine et la sortie du terminal doivent être des variables numériques, mais le processus intermédiaire est un processus d'inférence floue.

La raison pour laquelle il faut changer une variable numérique en une variable floue est que, du point de vue du contrôle flou ou de l'intuition d'un être humain, il n'existe pas de variables

absolument nettes dans notre monde réel.

L'avantage d'utiliser cette logique floue est qu'elle permet une interpolation fluide entre des variables linguistiques avec relativement peu de règles, fournit un moyen naturel de modéliser certains types d'expertise humaine dans un programme informatique, utilisant un langage imprécis et intrinsèquement robuste. En plus de cela, il a la capacité de modéliser les problèmes d'affaires très complexes impliquant de multiples experts.

Chapitre 4

Agent et ses spécificités

4.1 Introduction

Le développement de systèmes distribués [73] est influencé par plusieurs paradigmes. En outre, des technologies telles que les services Web sont maintenant considérés comme la norme, déployée dans les outils de développement communs et largement utilisé. Cependant, malgré cette tendance récente, le nombre sans cesse croissant de puissants dispositifs personnels va inévitablement relancer l'intérêt dans un autre paradigme connu sous le nom d'agents autonomes. Les agents sont en effet considérés comme l'un des principaux éléments constitutifs de l'infrastructure Web de la prochaine génération émergente. Les services Web sont des ressources très importantes pour les agents. Les agents doivent être en mesure de récupérer, d'exécuter et de composer des services Web, fournissant un support intelligent et personnalisé pour les utilisateurs. D'autre part, les agents doivent également être en mesure d'exporter leurs fonctionnalités comme les services Web afin d'être pleinement intégrés dans le paradigme orienté Service.

Les systèmes multi-agents font référence aux approches distribuées fondées sur la définition de plusieurs entités autonomes et intelligentes appelées "agents". Les agents logiciels sont envisagés comme des entités logicielles autonomes dynamiques qui agissent pour le compte de ses utilisateurs en fonction d'une donnée d'ordre du jour de buts.

La notion d'agent se réfère à une entité autonome évoluant en interaction avec son environnement, souvent dynamique et imprévisible.

La modélisation de ces agents et leurs interactions trouve son inspiration dans l'observation de systèmes biologiques complexes (sociétés animales organisés comme les colonies de fourmis, les nuées d'oiseaux ou les bancs de poissons)[74].

L'utilisation des SMA ne se limite pourtant pas à la simulation de vie artificielle mais ils sont aussi un outil très performant dans la résolution de problèmes complexes [75]. Les systèmes multi-agents sont composés d'un ensemble d'agents autonomes en interaction les uns avec les autres, par l'intermédiaire de l'environnement qu'ils partagent. L'intérêt porté aux SMA réside dans leur capacité à générer des comportements émergents (à travers les phénomènes comme l'auto-organisation) à partir des interactions individuelles.

4.2 Agent et Système multi-agents

Les systèmes multi-agents constituent un des axes de l'intelligence artificielle distribuée dans laquelle les systèmes permettent aux agents de coopérer et de coordonner leurs buts et leurs plans d'actions pour résoudre un problème [76].

Généralement, les systèmes multi-agents sont conçues pour modéliser les systèmes complexes correspondant à des ensembles constitués d'un grand nombre d'entités en interaction et situées dans un certain environnement[33]. Dans ce qui suit, nous proposons le contexte théorique des agents et systèmes multi agents :

Concept d'agent

— Un agent est un système informatique situé dans un environnement, il agit d'une façon autonome et flexible pour atteindre les objectifs pour lesquels il a été conçu [75].

— Un agent est une entité intelligente, agissant rationnellement et intentionnellement, en fonction de ses buts propres et de l'état actuel de sa connaissance[82].

— Les agents sont des entités logicielles qui interagissent avec un environnement et sont susceptibles de se modifier et d'évoluer en fonction des stimuli externes et internes, ces derniers étant dus aux capacités proactives et délibératives des agents eux-mêmes[83].

— Un agent est une entité active autonome agissant par délégation pour le compte d'un client[84].

Ferber [85] définit un agent comme étant une entité physique ou virtuelle :

— qui est capable d'agir dans un environnement, — qui peut communiquer directement avec d'autres agents, — qui est animée par un ensemble de tendances (objectifs, satisfaction, etc.),

— qui possède ses propres ressources,

— qui est dotée d'une capacité de perception de son environnement,

— qui n'a qu'une connaissance partielle et locale de son environnement,

— qui possède des compétences et peut offrir des services,

— qui peut éventuellement se reproduire,

— dont le comportement tend à satisfaire ses objectifs, en prenant en compte les ressources et compétences dont elle dispose, en fonction de sa perception, de ses représentations et des communications qu'elle reçoit.

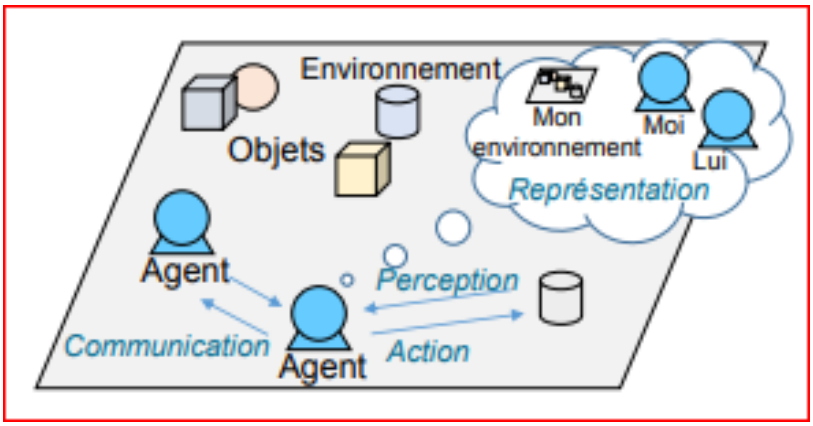

FIGURE 4.1 – Environnement des agents[74]

Pour Wooldridge et Jennings[86], le terme agent est utilisé pour décrire un matériel ou (plus couramment) un logiciel possédant les propriétés suivantes :

— **Autonomie :** les agents opèrent sans intervention extérieure ;

— **Capacités sociales :** les agents interagissent entre eux et éventuellement avec des utilisateurs humains ;

— **Réactivité :** les agents perçoivent leur environnement et répondent aux changements de celui-ci par des actions spécifiques ;

— **Pro-activité :** les agents sont capables de prendre des initiatives.

4.3 Caractéristiques d'un agent

Un agent présente un certain nombre de propriétés, dont certaines sont plus importantes que d'autres selon le type d'application. Voici les principales propriétés[86][87] :

1. **Situation :** l'agent est capable de recevoir des entrées sensorielles de son environnement et peut effectuer des actions qui changent cet environnement (par exemple : systèmes de contrôle de processus, démons logiciels).
2. **Autonomie :** l'agent est capable de prendre des décisions et d'agir sur son état interne sans l'intervention directe d'un tiers (humain ou un autre agent) et contrôle ses propres actions ainsi que son propre état interne.
3. **Réactivité :** l'agent est capable de percevoir son environnement et de répondre dans le temps requis aux changements qui apparaissent dans cet environnement.
4. **Pro-activité :** l'agent ne doit pas agir seulement en réponse à son environnement, mais il doit exhiber un comportement orienté but et opportuniste avec la capacité de prendre l'initiative au bon moment.
5. **Sociabilité :** l'agent doit être capable d'interagir, quand la situation l'exige, avec les autres agents et les humains via une sorte de langage de communication, afin d'accomplir ses tâches ou aider les autres dans leurs activités.
6. **Mobilité :** fonctionnalité supplémentaire, elle permet à l'agent de se déplacer de manière autonome à travers le réseau.

4.4 Différents types d'agents

En général, les SMA sont divisés en deux principales familles : les agents cognitifs et les agents réactifs.

Agents cognitifs :

les agents cognitifs interagisent selon leur structure et s'articulent autour de trois fonctions principales : percevoir, décider et agir. Chaque agent est spécialisé dans un domaine et sait com-

muniquer avec les autres. Ils possèdent des buts et des plans explicites leur permettant d'accomplir leurs buts[88].

— Représentation explicite de soi, environnement et les autres agents.

— Organisation explicite.

— Interaction explicite et élaborée.

Agents réactifs :

Les agents réactifs sont des agents qui réagissent par stimulus-réponse à l'état courant de l'environnement[88]. Ils se sont des agents sans intelligence(sans anticipation, sans planification). Des comportements intelligents peuvent émerger de leur association.

— Pas de représentation explicite.

— Organisation implicite/induite.

— Communication via l'environnement.

Agents cognitifs	Agents réactifs
Représentation explicite de l'environnement	Pas de représentation explicite
Peut tenir compte de son passé	Pas de mémoire locale
Agents complexes	Fonctionnement stimulus/réponse
Nombre d'agents réduit	Nombre d'agents élevé

TABLE 4.1 – Agent cognitif et agent réactif[88]

Agents hybrides

Il est possible de concevoir une architecture d'agent qui combine entre les architectures citées précédemment (cognitif et réactif), on parlera alors des agents hybrides. La majorité des modèles d'agents hybrides proposent de décomposer chaque agent en différents modules réactifs et cognitifs avec un module spécifique qui l'activation des autres modules[89].

Dans la littérature, il y a d'autres types d'agents, nous pouvons citer :

Agent adaptatif

Un agent adaptatif est un agent qui est capable de contrôler ses aptitudes (de communication, comportementales, etc.) selon l'agent avec lequel il interagit[88].

Agent Intelligent

On appelle agent intelligent, un agent cognitif, rationnel, intentionnel et adaptatif[88].

Agent intentionnel :

Un agent intentionnel est un agent guidé par ses buts [88]. Il possède des buts et des plans explicites lui permettant d'accomplir ces buts[85].

Agent rationnel

Un agent rationnel est un agent dont les actions sont toujours le fruit d'une délibération raisonnée et qui servent directement à la satisfaction des buts de l'agent [85].

Agent mobile

Avec le progrès de l'utilisation de l'internet et notamment des services web, Le concept agent mobile a connu un intérêt croissant ces dernières années, La notion d'agent mobile invoque deux concepts : l'agent et la mobilité.

Le concept d'agent se réfère au domaine de l'Intelligence Artificielle (IA) alors que celui de la mobilité des agents repose sur les caractéristiques empruntées à la mobilité du code[90].

4.5 Système multi agents

Selon Ferber[85], on appelle système multi-agent, un système composé des éléments suivants :

1. Un environnement E, c'est à dire un espace disposant généralement d'une métrique.

2. Un ensemble d'objets O. Ces objets sont situés, c'est à dire que, pour tout objet, il est possible à un moment donné, d'associer une position dans E. Ces objets sont passifs, c'est à dire qu'ils peuvent être perçus, crées, détruits et modifiés par les agents.
3. Un ensemble A d'agents, qui représentent les entités actives du système.
4. Un ensemble de relations R qui unissent les objets et les agents entre eux.
5. Un ensemble d'opérations Op permettant aux agents A de percevoir, produire, consommer, transformer et manipuler les objets de O.
6. Des opérateurs chargés de représenter l'application de ces opérations et la réaction du monde à cette tentative de modification, que l'on appellera les lois de l'univers.

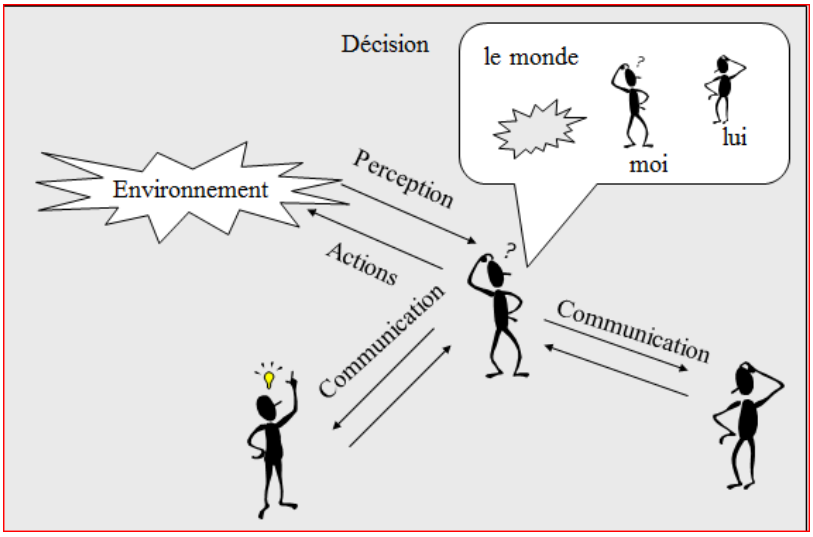

FIGURE 4.2 – Environnement des agents[89]

4.6 Communication inter-agents

Les agents communiquent et interagissent pour synchroniser leurs actions et pour résoudre leurs conflits, ils communiquent également pour s'aider mutuellement. Généralement, les agents communiquent lorsqu'ils sont face à un problème qui ne savent pas résoudre, lorsqu'il est nécessaire de coordonner leurs actions ou encore lorsqu'il y a un conflit entre plusieurs agents et que le conflit ne peut pas être résolu d'une manière déterministe.

La communication entre agents se fait de deux manières : la communication par envoi de message et la communication par partage d'informations.

Communication par envoi de messages

Les systèmes multi-agents fondés sur la communication par envoi de messages se caractérisent par le fait que chaque agent possède une représentation propre et locale de l'environnement qui l'entoure. Chaque agent interroge les autres agents sur cet environnement ou leur envoyer des informations sur sa propre perception des choses.

Chaque message contient une continuation, à qui doit être retournée la réponse à la requête. Il y a distribution des connaissances (chaque acteur possède un comportement réparti entre ses accointances) et distribution du contrôle (chaque acteur possède un script qui définit sa réaction aux messages qu'il reçoit)[89]. La communication se fait soit en mode point à point, soit en mode par diffusion

Communication par partage d'information

La communication entre les différents agents du système est réalisée par partage d'information lorsque ceux-ci disposent d'une zone de mémoire commune. Cette mémoire partagée est vue comme un tableau sur lequel les agents écrivent, trouvent des réponses partielles, des informations. Le tableau noir est divisé en niveaux. Les agents travaillant à un niveau particulier d'abstraction ont accès à un niveau correspondant dans le tableau. Un dispositif de contrôle gère les conflits d'accès au tableau, les agents faisant les demandes d'accès de manière autonome[91].

4.7 Agent et service Web

Les agents logiciels ont été envisagés comme utilisateur potentiel de services Web sémantiques afin d'interagir avec des descriptions sémantiques des services Web pour découvrir de manière autonome, sélectionner, composer, invoquer et exécuter les services basés sur les besoins des utilisateurs. Cependant, il existe un déficit de communication entre les deux. La principale raison est que les agents logiciels ne sont pas compatibles avec les normes largement

acceptées de services Web. L'utilisation de la notion d'agent pour les services web est ainsi un enjeu de taille pour doter les services web des capacités intéressante des agents logiciels.

Les agents étendent les services Web de plusieurs manières importantes [92] :

— Un service Web ne se connaît que lui-même, mais pas ses utilisateurs, ses clients. Les agents sont souvent conscients de soi et des autres agents et de leurs capacités à mesure que des interactions entre agents se produisent.

— Contrairement aux agents, les services Web ne sont pas conçus pour utiliser et réconcilier des ontologies. Si le client et le fournisseur d'un service utilisent des ontologies différentes, le résultat de l'appel de service Web serait incompréhensible pour le client.

— Les agents sont intrinsèquement communicatifs, tandis que les services Web sont passifs jusqu'à ce qu'ils soient invoqués. Les agents peuvent fournir des alertes et des mises à jour lorsque de nouvelles informations sont disponibles.

— Un service Web, tel qu'il est actuellement défini et utilisé, n'est pas autonome. L'autonomie est une caractéristique des agents. Parmi les agents, l'autonomie se réfère généralement à l'autonomie sociale, où un agent est conscient de ses collègues et est sociable, mais exerce son indépendance dans certaines circonstances.

— Les agents sont coopératifs et, en formant des équipes, les coalitions peuvent fournir des services plus complets.

— Les agents peuvent fournir des services à d'autres agents, mais aussi aux consommateurs de services Web, qu'ils soient commerciaux ou individuels.

— Les agents peuvent obtenir des informations du grand référentiel sur le Web sous la forme de services Web.

Nous pouvons également citer d'autres avantages des agents logiciels cités dans [93] :

— Les agents permettent d'effectuer une tâche même lorsqu'ils sont déconnectés en raison de leur asymétrie et de l'autonomie ;

— Les agents peuvent communiquer et coopérer les uns avec les autres, accélérant et facilitant le processus de découverte.

Le tableau suivant donne une comparaison entre le concept agent et le concept service web[33] :

Agent	Service Web
Un agent est autonome, a une connaissance sur lui-même et de ses accointances.	Un service web a des connaissances sur lui-même, mais pas sur ses utilisateurs, clients.
Les agents sont communicatifs.	Les services Web sont passifs jusqu'à ce qu'ils soient invoqués.
Les agents sont conçus pour utiliser et régénérer des ontologies.	Les services Web ne sont pas conçus pour utiliser et régénérer des ontologies.
Les agents peuvent être conçus pour être coopératifs, ce qui nous permet d'effectuer une composition de services compréhensible en utilisant différents protocoles.	Les services Web ne fournissent pas des protocoles de coopération qui peuvent conduire à une composition de services.
Les agents peuvent s'adapter à de nouveaux environnements.	Les services Web ne prennent pas en charge l'adaptabilité.

TABLE 4.2 – Comparaison entre agent et service Web[33].

4.8 Conclusion

À travers ce chapitre, nous avons essayé de faire un survol sur le paradigme agent qui a comme atout la distribution qui permet de contrôler la complexité des problèmes à résoudre en décomposant par exemple le traitement. Comme conclusion de ce chapitre, nous pouvons dire que, les services Web sont des ressources très importantes pour les agents. Les agents doivent être en mesure de récupérer, d'exécuter et de composer des services Web, fournissant un support intelligent et personnalisé pour les utilisateurs. D'autre part, les agents doivent également être en mesure d'exporter leurs fonctionnalités comme les services Web afin d'être pleinement intégrés dans le paradigme orienté Service.

Chapitre 5

Travaux Connexes

5.1 Introduction

Après la présentation du contexte du travail ainsi que les concepts reliés au domaine de notre recherche, dans le présent chapitre, nous présentons quelques travaux qui ont contribués pour résoudre la problématique de la prise en charge des données linguistiques lors de la sélection d'un service Web et ainsi de donner plus de liberté et de choix aux clients du Web.

5.2 Découverte des services Web

La sélection d'un service approprié en fonction des besoins du client est une tâche complexe, car il existe un plus grand nombre de services Web similaires disponibles pour satisfaire une tâche particulière.

Avant de sélectionner un service Web pour le rendre au client pour qu'il procède à son invocation, il est primordial de passer par l'étape de la découverte. La découverte est définie comme étant la localisation automatique des services répondant à une requête utilisateur[100].

Un processus de découverte de service Web est effectué en trois étapes principales. La première étape est la publicité du service Web par les développeurs. La deuxième étape est la demande de service Web par l'utilisateur. L'utilisateur envoie une demande de service Web spécifiant l'exigence dans un format prédéfini au référentiel de service Web. Le service Web Matcher, qui fait partie intégrante du modèle de découverte de service Web, fait correspondre la requête

de l'utilisateur avec les services Web disponibles et trouve un ensemble de candidats au service Web. L'étape finale est la sélection et l'invocation de l'un des services Web récupérés[11].

Plusieurs critères peuvent être utilisés pour localiser les services Web similaires, parmi ces techniques, nous pouvons citer la technique de matching[101].

Les algorithmes de matching consistent à comparer les données ainsi que l'algorithme d'appariement de la requête avec la description du service.

Généralement,On distingue les approches fonctionnelles et les approches non fonctionnelles :

Les approches non fonctionnelles comparent les valeurs des parametres des QoS des services Web avec la requête de l'utilisateur et les classent. Les algorithmes de comparaison sont généralement des méthodes de décision multicritères comme TOPSIS[102] et Electre[103].

Les approches fonctionnelles comparent les descriptions d'interfaces ou de comportement des services Web, elles peuvent être syntaxiques (WSDL), sémantiques (OWLS, SAWSDL, WSMO) ou comportementales.

5.3 Découverte des services Web à base de QoS

La majorité des travaux sur la découverte à base des QoS offrent aux fournisseurs de services la possibilité de publier des informations sur la qualité de service alors que dans[105] , certains défis doivent être pris en compte. Ces défis comprennent :

- automatiser, administrer et maintenir les informations de qualité de service mises à jour dans les annuaires ;
- garantir la validité des informations de QoS fournies par les fournisseurs de services ;
- effectuer des mesures QoS de manière ouverte et transparente ;
- gérer le format des résultats de l'information sur la qualité de service ;
- améliorer l'UDDI pour soutenir les informations de qualité de service avec la version existante sans nécessiter de modifications ou d'extensions des spécifications.

Afin de fournir un classement axé sur la qualité des services Web, il est important de recueillir des informations sur la qualité de service concernant les services Web. En plus, il faut fournir un moyen chargé de mesurer les informations de QoS pour les services Web collectés.Dans ce

travail, les clients peuvent choisir et gérer sélectivement leurs critères de recherche via une interface utilisateur graphique. Une fois que les clients soumettent leurs demandes de recherche, une fonction de pertinence de service Web est utilisée pour mesurer le classement de pertinence d'un service Web donné.

Dans[6], les auteurs ont proposé une architecture dans laquelle le registre UDDI prend en charge l'intégration des paramètres QoS avec les informations de services Web, qui se résume comme suit : Une fois le service Web publié, les informations sur la qualité des services sont stockées dans le registre UDDI, c'est-à-dire les informations de QoS peuvent également être stockées dans les répertoires UDDI avec les informations de service Web. Un négociateur de services ajouté à ce modèle, ce négociateur aide le client à choisir le service Web approprié en fonction des paramètres QoS. Le fournisseur de services assure l'enregistrement, la suppression et la mise à jour des services Web. Le service de publication publie des informations de fonctionnalité de service dans le registre UDDI après le processus d'authentification QoS. Le consommateur peut effectuer une recherche dans le registre du service Web via le négociateur de services Web. Il envoie sa requête qui inclut ses besoins fonctionnels ainsi que ses préférences en termes de QoS. Il choisit le service Web qui satisfait les préférences utilisateur et les contraintes de QoS du registre. Le selectionneur du service Web prend également la consultation du gestionnaire de réputation pour connaître le score du service. L'approche présentée dans cet article n'inclut pas la logique floue, elle ne prend donc pas en compte les informations des consommateurs imprécises et incertaines. Pour tester ce modèle, les auteurs ont utilisé les critères de qualité suivants : temps de réponse, sécurité, fiabilité et coût. Ainsi, pour classer les services Web similaires, ils ont opté pour les méthodes de classement AHP et TOPSIS.

5.4 Evaluation des qualité de services

Dans[20], l'auteur a proposé l'extension de l'architecture conventionnelle SOA pour qu'elle supporte le QoS Bootstrapping en ajoutant un module Evaluation de la Qualité de Service à la Publication des Services Web. Le Bootstrapping QoS est le processus d'évaluation de la QoS des services nouvellement enregistrés au moment de la publication des services.

Dans cet article, une solution pour la QoS bootstrapping est introduite qui évalue les at-

tributs QoS pour les services Web nouvellement enregistrés. La contribution principale réside dans l'approche automatisée pour l'amorçage QoS au moment de la publication et avant que tout demandeur ne demande le service Web. Par conséquent, la justification de la qualité de service pour les nouveaux services Web sera disponible au moment de la publication des services Web, à savoir :

— il n'est pas nécessaire de tester les services Web au moment de la recherche du service ;
— l'opération de recherche sera plus rapide ;
— augmente la possibilité pour les nouveaux services Web d'être sélectionné par les demandeurs ;
— augmente le niveau de confiance dans ces services.

Il considère que les critères non fonctionnels sont pris en compte dans la sélection des services Web et propose un cadre permettant l'extension de l'architecture. Ce cadre améliore la satisfaction des consommateurs pour la sélection des services Web puisque la sélection est prise en charge en plus des critères fonctionnels sur des critères non fonctionnels tels que temps de réponse, latence, fiabilité et accessibilité qui feront la différence et offriront le meilleur service au consommateur.

5.5 Méthodes de prise de décision multi-critères Floue

Dans[106], comme les critères pris en compte dans un problème de prise de décision sont très vagues et incertains, les auteurs décrivent la conception d'un système d'aide à la décision floue dans une approche d'analyse multicritères pour sélectionner les meilleures alternatives. La méthode du processus de hiérarchie analytique floue (FAHP) est utilisée pour déterminer les pondérations préférentielles des critères pour les décideurs par perception subjective (langage naturel).

Dans[99], en répondant à la problématique suivante : les approches classiques de classification sont difficiles à appliquer dans la pratique en raison d'informations incomplètes et non quantifiables et d'un jugement humain imprécis. L'auteur propose une approche de décision de groupe multi-critères utilisant la logique floue et la méthode Entropy pour le calcul des poids par rapport aux critères qui ont une grande influence sur le résultat final de la sélection. Il pré-

tend étendre la méthode TOPSIS pour soutenir les données linguistiques des consommateurs. L'algorithme proposé se compose de 6 étapes de la matrice de décision formées par la linguistique passant par la normalisation, calcul des poids par rapport aux critères utilisant la méthode, construction Entropy, PIS et NIS, détermination de la proximité du coefficient et fin de la classification des préférences. Pour démontrer la faisabilité de son approche, l'auteur a testé son algorithme avec trois alternatives dont chacune est caractérisée par quatre 4 critères.

5.6 Découverte à base de la logique floue

Dans[96],les auteurs ont proposé un modèle permettant de décrire les propriétés QoS des services Web en termes flous et, à la fin, de trouver la valeur numérique pour le classement des services Web.

Pour classer les services en fonction de leur valeur QoS, développer un algorithme de hiérarchisation "PROMETHEE modifié". Les modules suivants ont été inclus dans le système.

— Demandeur de service : le demandeur de service est celui qui demande un service. Le demandeur de service peut entrer les exigences fonctionnelles et non fonctionnelles des services souhaités à la fois dans le terme flou ou crisp ;

— Fournisseur de services : le fournisseur de services est celui qui met en œuvre les services et les publie dans le registre de service avec la description du service pour un usage public. Le fournisseur de services peut saisir la description fonctionnelle et non fonctionnelle de nouveaux services Web à la fois dans un terme flou ou numérique ;

— Registre de service : Le registre de service est la base de données qui contient des informations sur les services tels que l'URL des services (URL du fichier .owl). Ces URL selon les besoins des utilisateurs sont récupérés à partir de la base de données du service pour répondre aux besoins de l'utilisateur ;

— Agent de publication du service : le fournisseur de services peut demander l'enregistrement d'un nouveau service via l'agent de service de publication. Cet agent conservera la description du nouveau service Web et de ses informations de QoS dans sa base de données.

Dans[107],les auteurs proposent un algorithme sur 7 étapes basé sur la logique floue pour

traiter le problème (prise de décision multi-attributs MADM). Son but est de trouver la meilleure alternative parmi plusieurs alternatives existantes. Les données relatives aux critères sont basées sur l'incertitude, pour cela l'auteur a utilisé la représentation des nombres triangulaires flous pour représenter les données linguistiques. Et comme le poids des critères a une grande influence sur la prise de décision, l'auteur a proposé la méthode Entropy pour la détermination des poids à l'étape 3 de l'algorithme et ensuite réussi le calcul de PIS et FIS et le coefficient de proximité les alternatives. L'auteur a testé son algorithme par trois alternatives ou chacune est caractérisée par quatre critères.

5.7 Agents et logique floue

Dans[108], les auteurs expliquent comment les agents intelligents et la logique floue peuvent contribuer à augmenter la qualité et la quantité d'un jeu vidéo. Ils ont appliqué cette approche dans le jeu BattleCity Game dont lea agent étaient de type BDI("Croyance-Désir-Intention", en anglais "Belief-Desire-Intention"). L'approche orientée agent du développement de jeux offre de nombreux avantages tout au long du cycle de développement. Il fournit une façon naturelle de modéliser les créatures du jeu et une architecture logicielle de haute flexibilité et de faible couplage qui permet l'épanouissement des comportements. La logique floue est une autre technique intelligente qui pourrait être utilisée pour booster la performance des jeux.

Dans[109], les auteurs proposent une approche (figure5.1) basée agent et logique floue pour la surveillance du réseau. D'une part, il utilise des règles pour définir des actions simples, basées sur une simple condition et une description d'action. D'autre part, la connaissance principale de cette solution est définie par la logique floue.

Ce système est capable de gérer et de modifier ses connaissances pour mieux s'adapter aux ressources surveillées. La connaissance de ce concept est répartie entre tous les agents. Les agents résidant sur des hôtes différents gèrent leurs parties de connaissances et sont capables de les partager/échanger.

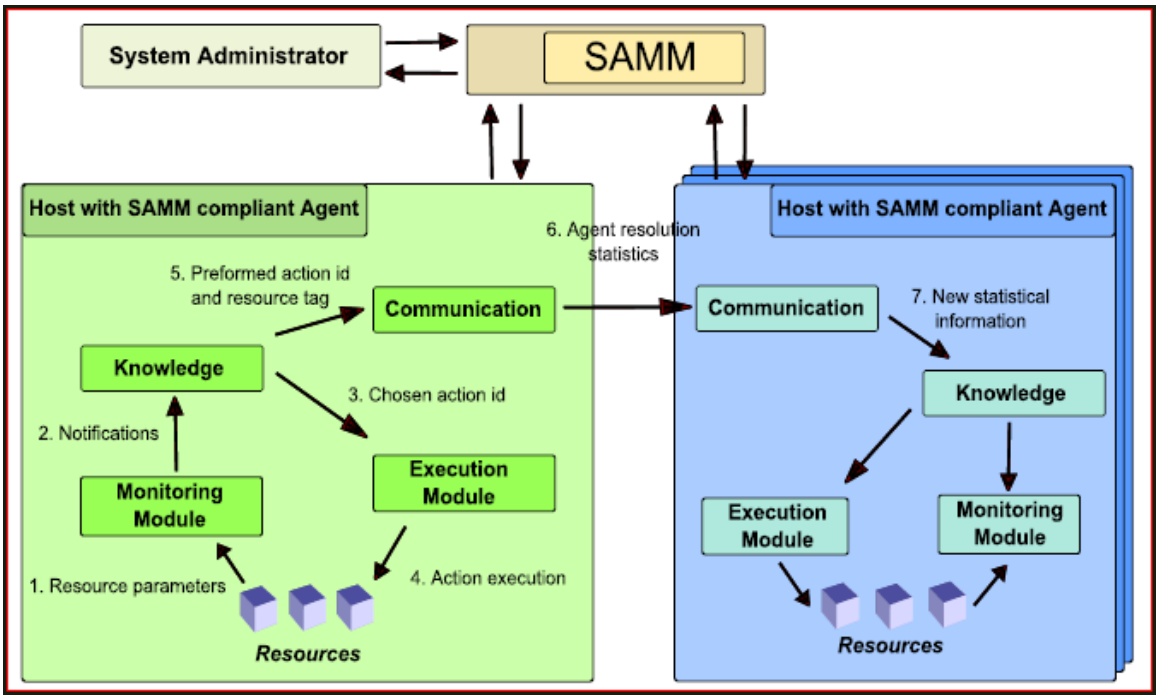

FIGURE 5.1 – Communication entre agents[109]

Dans[110], les auteurs proposent une nouvelle méthode pour l'extraction de l'information dans un ensemble de connaissances afin de répondre aux consultations de l'utilisateur en utilisant le langage naturel. Le système est basé sur un moteur Fuzzy Logic, qui tire parti de sa flexibilité pour gérer des ensembles de connaissances accumulées. Ces ensembles peuvent être construits dans des niveaux hiérarchiques par une structure arborescente. Le but de ce système est de concevoir et de mettre en place un agent intelligent pour gérer tout ensemble de connaissances où l'information est abondante, vague ou imprécise.

Dans[97], lea auteurs traitent de la planification distribuée dans un système multi-agent constitué de plusieurs agents intelligents, chacun devant interagir avec les autres agents autonomes.

Les auteurs proposent l'utilisation de la logique floue pour représenter la réponse de l'agent en cas d'interaction. Un agent est constitié de trois modules (figure 5.2) : Fuzzification, Fuzzy inference engine et defuzzification en plus d'une base de règles floues.

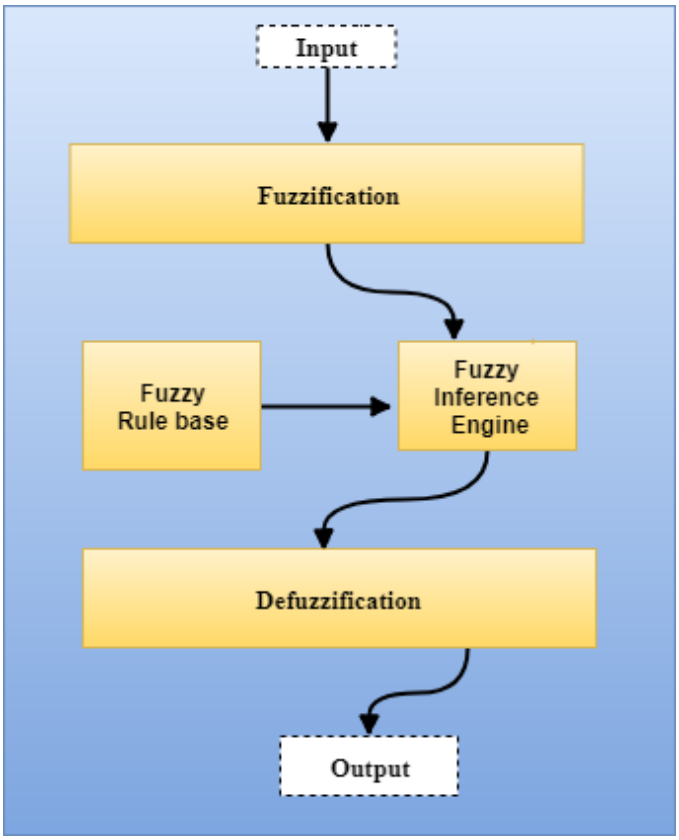

FIGURE 5.2 – Structure d'un agent[97]

5.8 Conclusion

Un service Web est défini comme étant un mécanisme qui permet d'accéder aux fonctionnalités, par le biais d'une interface avec des détails sur la façon dont l'accès est effectué. Le terme est également associé à un fournisseur de services, qui fournit les capacités, et à un ensemble de contraintes et de politiques. La qualité des modèles de services ont tendance à conduire à une représentation pauvre de la réalité, principalement en raison de leur manque de capacité à représenter l'imprécision. La logique floue fournit un cadre naturel pour faire face à l'incertitude et la tolérance aux données imprécises.

Les agents étendent les services Web de plusieurs manières importantes telles que : L'autonomie qui est une caractéristique des agents. Parmi les agents, l'autonomie se réfère généralement à l'autonomie sociale, où un agent est conscient de ses collègues et est sociable, mais exerce son indépendance dans certaines circonstances. en terme de communication, les agents

peuvent fournir des alertes et des mises à jour lorsque de nouvelles informations sont disponibles.les agents sont coopératifs et, en formant des équipes, les coalitions peuvent fournir des services plus complets. Encore Les agents peuvent fournir des services à d'autres agents, mais aussi aux consommateurs de services Web, qu'ils soient commerciaux ou individuels, aussi, Les agents peuvent obtenir des informations du grand référentiel sur le Web sous la forme de services Web.

Deuxième partie

Contributions

Chapitre 6

Approches Proposées

6.1 Introduction

Le nombre de services Web offrant la même fonctionnalité ne cesse pas de croître, donc la tâche de sélectionner un service Web parmi plusieurs services similaires est une tâche très délicate pour le consommateur de service car généralement, les services fournissant la fonctionnalité en question seront retournés dans l'ordre de leur inscription dans le registre UDDI en question[10]. Encore les approches de sélection tiennent compte du besoin du consommateur dans la sélection des services Web mais ne tiennent pas compte d'autres contraintes telles que les qualités fonctionnelles et non fonctionnelles[50].

Pour remédier à ce type de problème, il a été proposé de prendre en compte les critères de qualité de service et plus précisément lors du processus de sélection[20][66].

Là encore, l'évaluation du service Web par le consommateur n'est pas toujours précise et parfois ambiguë, d'où la nécessité de traduire cette évaluation en une logique analogue à la logique humaine et d'adopter des critères de qualité de service pour aider le consommateur à choisir ses services souhaités. Par conséquent, la logique floue peut être appliquée pour supporter une représentation imprécise des contraintes QoS[65] et présenter les préférences de l'utilisateur en tant que propriétés QoS avec une présentation floue car elles sont mieux adaptées à l'interprétation des termes linguistiques[67].

La sélection porte sur le choix d'une implémentation de service parmi celles qui sont découvertes pour la description donnée. La découverte est un prérequis pour la sélection, mais c'est

la sélection qui est le problème principal.

Dans la même veine, Dans ce chapitre nous allons présenter nos contributions pour apporter une solution à cette problématique.

Nous commençons dans un premier temps par présenter en détail nos deux contributions qui résolvent le problème de la prise des données linguistiques de l'utilisateur d'une part et d'autre part de sélectionner le service web approprié à retourner au demandeur.

La première contribution consiste en la proposition d'un nouveau modèle de sélection de services Web basé sur la logique floue. L'utilisation de la logique floue permet de représenter les données vagues ou inexactes en données avec des variables linguistiques et des valeurs floues.

Dans la deuxième contribution, nous allons présenter une approche de sélection des services Web basée sur les notions d'agent et logique floue. Les agents sont considérés comme l'un des principaux éléments constitutifs de la prochaine génération de l'infrastructure Web. Afin de satisfaire des clients avec des services répondant à leurs besoins en termes de qualité, nous optons d'intégrer le paradigme agent, L'utilisation de la notion d'agent pour les services web est donc un défi majeur pour équiper les services web avec des capacités intéressantes d'agents logiciels.

Les services Web sont des ressources très importantes pour les agents. Les agents doivent être en mesure de récupérer, d'exécuter et de composer des services Web, offrant un support intelligent et personnalisé aux utilisateurs. D'un autre côté, les agents doivent également être en mesure d'exporter leurs fonctionnalités en tant que services Web pour être pleinement intégrés dans le paradigme orienté services.

L'utilisation de la logique floue permet de représenter les données vagues ou inexactes en données avec des variables linguistiques.

6.1.1 Qualité de service :

La qualité de service[8] peut être définie comme une comparaison entre les attentes du client et ce qui est réellement offert. Elle est définie comme une combinaison de plusieurs critères qui peuvent alors être considérés comme un critère de choix lorsque l'on sélectionne parmi plusieurs services web découverts ceux qui respectent les contraintes imposées.

6.1.2 Paramètres de qualités de services :

La sélection des services web à base de Qualités de Services (QoS) consiste à choisir parmi les services Web découverts, ceux qui répondent mieux aux exigences de l'utilisateur sur la base des besoins non fonctionnels QoS.

Il existe des mésures pour quantifier un service Web. Ces dernieres sont présentés par la qualité de service. De plus, les QoS représentent l'aspect non-fonctionnel d'un service. Ces valeurs permettent la sélection des services pertinents aux demandes des utilisateurs.

Plusieurs critères de qualité de service sont décrits dans différents travaux et les propriétés de QoS les plus représentatives sont présentées comme suit :

- **Disponibilité :** la probabilité que les ressources, les services sont disponibles pour les parties autorisées en tout temps pour l'utilisation ou le pourcentage de temps que le service fonctionne [8].
- **Fiabilité :** La capacité d'un service à exécuter les fonctions requises dans des conditions spécifiées pendant une période donnée [68].
- **Débit (Throughput) :** Le nombre de demandes de service terminées sur une période donnée[69].
- **Temps de réponse :** la durée entre l'envoi d'une demande par un utilisateur de service et la réception d'une réponse [70].
- **Sécurité :** L'aspect qualité du service Web pour assurer la confidentialité et la non-répudiation en authentifiant les parties impliquées, en chiffrant les messages et en assurant le contrôle d'accès. Le fournisseur de services peut avoir différentes approches de sécurité en fonction du demandeur de service[71].
- **Accessibilité :** c'est l'aspect qualité d'un service qui représente le degré de capacité à servir une requête de service Web. Il peut être exprimé comme une mesure de probabilité indiquant le taux de réussite ou la probabilité d'une instanciation de service réussie à un moment donné [71].
- **Coût d'exécution :** Le coût à dépenser pour exécuter un service Web, ce coût peut être fourni par le fournisseur de services[66].
- **Réputation :** Le taux moyen du service rapporté par le client, il est basé sur le service client; cela dépend principalement de l'expérience de l'utilisateur final de l'utilisation

du service. Différents utilisateurs peuvent avoir des opinions divergentes sur le même service, généralement ; le client évalue le service en le classant avec une valeur[72].
Dans les approches proposées, nous allons opté pour l'utilisation de ces critères non fonctionnels de services Web dans le processus de sélection.

6.2 Un modèle de sélection des services Web basée sur la logique floue

Cette contribution consiste en un nouveau modèle de sélection de services Web basé sur la logique floue. L'utilisation de la logique floue permet de représenter les données vagues ou inexactes en données avec des variables linguistiques et des valeurs floues. Notons que parmi les intérêts de la logique floue est de formaliser le raisonnement humain est que les règles sont énoncées en langage naturel [48].

L'utilisation de la logique floue dans notre approche est primordiale, avec cette logique, nous présentons les valeurs de critères de qualité de service relatives aux services Web en valeurs linguistiques : par exemple au lieu de donner une seule valeur numérique au paramètre C_1, on lui attribue par exemple : mauvais, moyen, bon et excellent.

6.2.1 Architecture du modèle proposé

Le modèle proposé [49] (figure suivante) est décrit comme suit : Initialement, l'utilisateur envoie sa requête fonctionnelle au registre UDDI qui contient le nom du service Web suhaité et leurs paramètres QoS. En réponse à cette demande et après l'étape de la découverte, une liste de services similaires est produite. L'extraction de QoS est effectuée pour chaque service Web afin de produire une matrice de services Web flous et leur QoS correspondante. Une fois que la matrice floue est prête, elle est soumise au moteur d'inférence floue qui se chargera d'appliquer les règles d'inférence floues pour produire un score pour chaque service Web. À la fin, ces valeurs de score sont classées pour fournir le meilleur service à recommander au client.

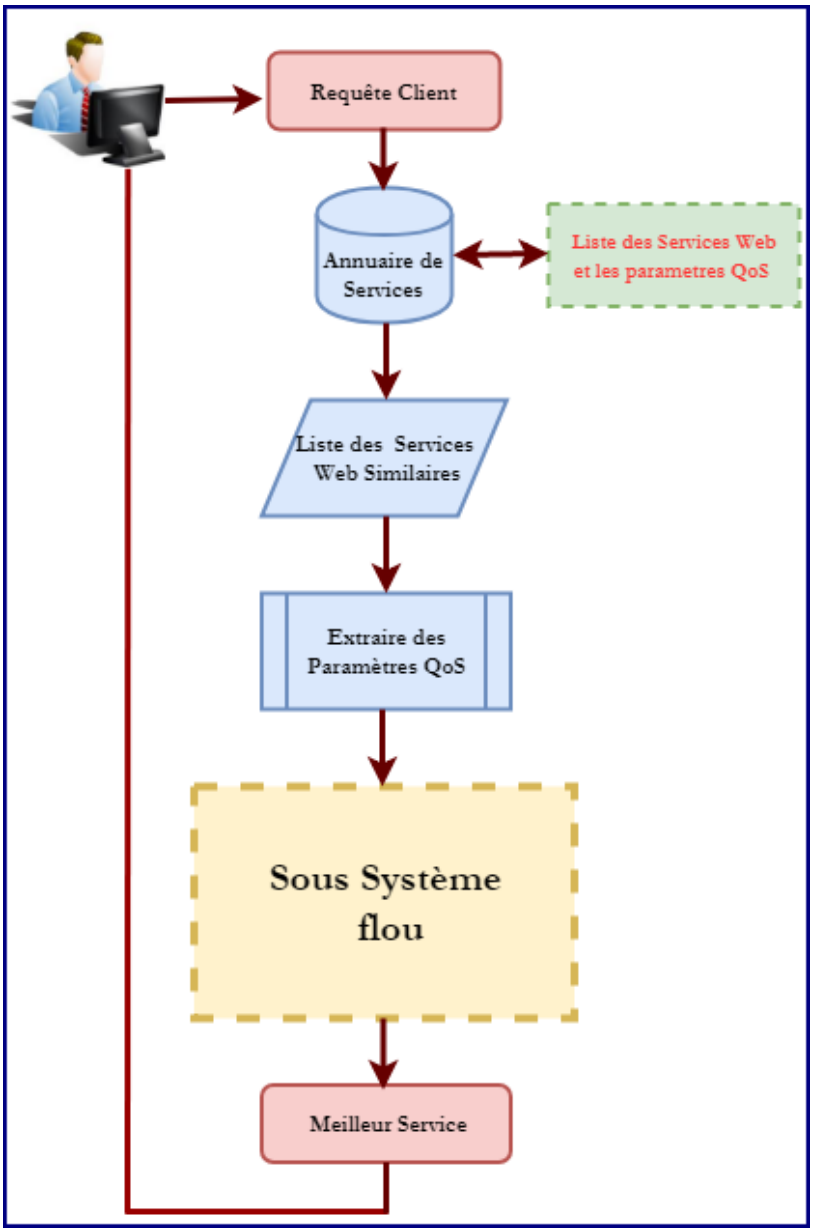

FIGURE 6.1 – Architecture du modèle proposé

Algorithme de sélection flou

L'algorithme de sélection de services Web similaires est représenté comme suit :

Entrée : la requête contient des paramètres fonctionnels.

Sortie : Services Web classés

1. Obtenir la requête de l'utilisateur
2. Trouver la liste de la recherche des services similaires à partir du registre UDDI.

3. Extraire les propriétés non fonctionnelles (QoS) du service.
4. Passer chaque QoS au sous-système flou
5. En choisissant un type de fonction d'appartenance (par exemple, méthode d'appartenance triangulaire, trapézoïdale ou gaussienne, traduire les valeurs croisées des paramètres QoS en valeurs floues.
6. Pour chaque service Web, le moteur d'inférence flou calcule le score approprié en utilisant la base de règles floues.
7. Defuzzifier les valeurs de chaque service Web en utilisant la moyenne des maxima ou méthode de centre de gravité pour obtenir une valeur numérique nette de chaque service Web. Cette valeur va par la suite qualifier le service Web.
8. Classement des services obtenus.

Diagramme de séquence de sélection

L'ensemble des interactions entre les différents acteurs est présenté avec le diagramme de séquence suivant :

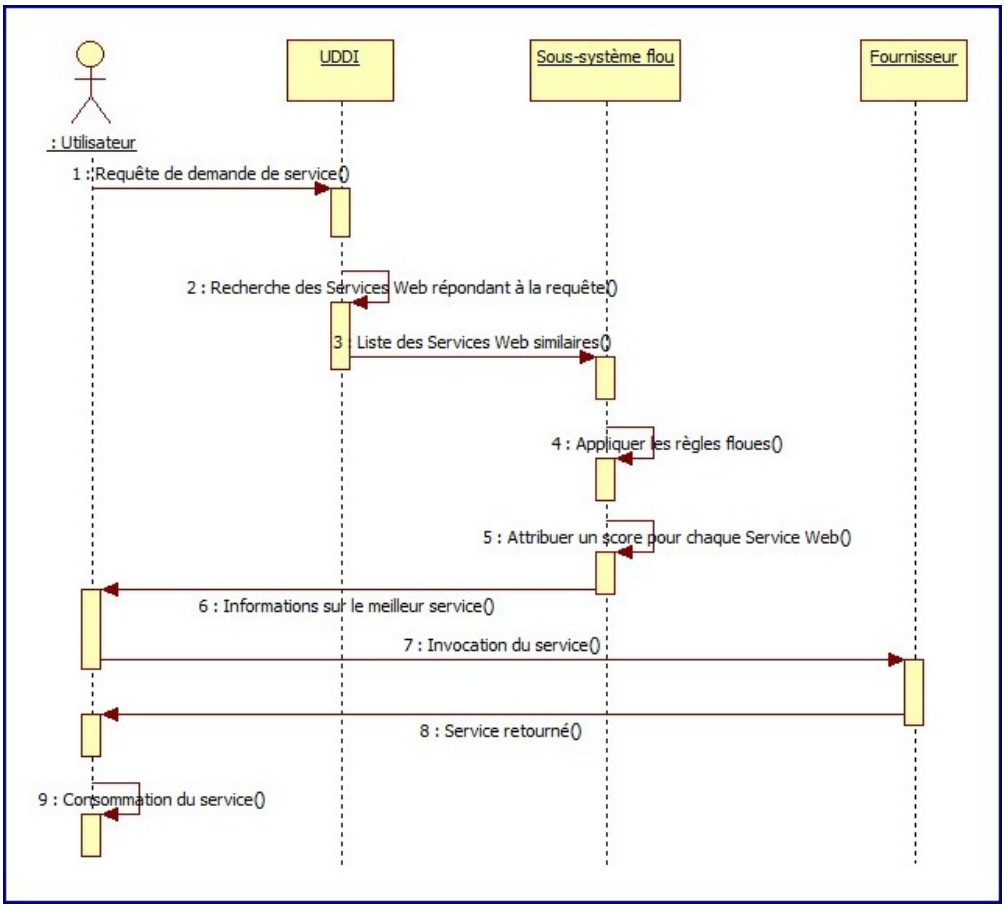

FIGURE 6.2 – Diagramme de séquence : Processus de sélection

Sous-Système Flou

Ce sous-système est décrit comme suit (figure 6.2) : Il a comme entrée les valeurs quantitatives des paramètres de qualité des services Web. Il construit une matrice qui a des services Web en ligne et des colonnes les critères de qualité appropriés. Ces valeurs numériques sont passées au module de Fuzzification qui, avec l'aide d'experts humains et avec le choix d'une fonction d'appartenance donnée, procède à la fuzzification pour donner comme résultat une matrice de décision floue.

Ensuite, cette matrice est reçue par le moteur d'inférence floue qui, avec l'exploitation de la base de données de règles floues produite pour chaque service Web, aboutit à un terme flou nommé score flou.

Ce score flou est redirigé vers le module de défuzzification qui, avec l'utilisation d'un pro-

cédé de défuzzification, produit une valeur quantitative caractérisant le service Web en question.

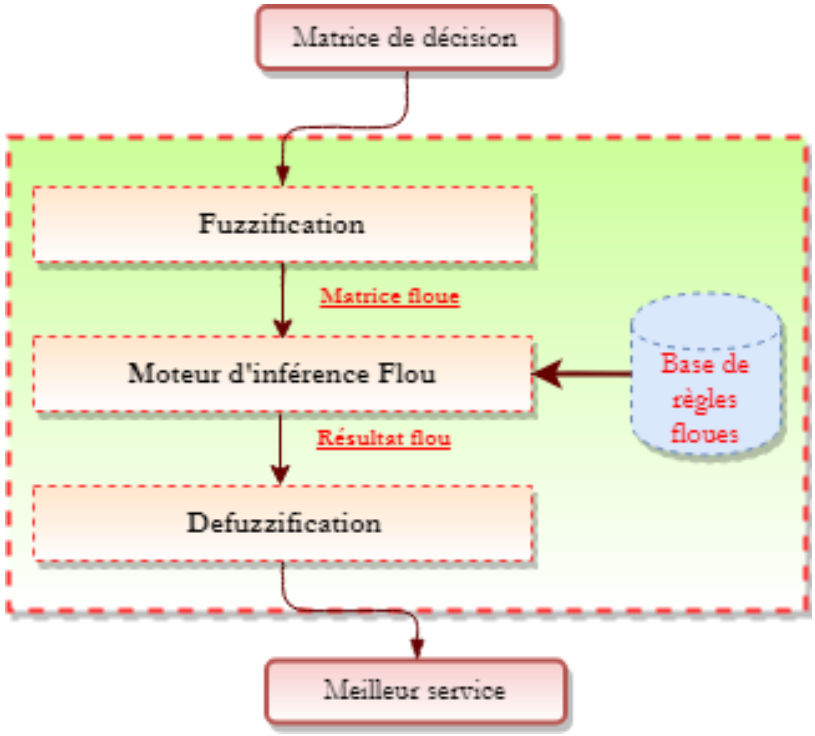

FIGURE 6.3 – Sous-Système Flou

Fuzzification

Le but de l'étape de fuzzification est de transformer une donnée numérique en une variable linguistique. Pour cela, nous devons créer des fonctions d'appartenance qui définissent le degré d'appartenance d'une donnée numérique à une variable linguistique. Les entrées de la fuzzification sont les paramètres de qualité de service (QoS) dont les valeurs sont quantitatives.

Pour chaque critère, nous pouvons créer plusieurs fonctions d'appartenance à savoir : Fonction triangulaire, trapézoïdale ou gaussienne. Si les services Web que nous voulons tester sont publiés avec les critères : fiabilité, disponibilité et temps de réponse : chacune de ces données aura plusieurs fonctions d'appartenance. Par exemple : Si nous voulons transformer la fiabilité en variable linguistique. Nous pouvons trouver plusieurs valeurs linguistiques qualifiant ces données numériques : faible, moyenne et élevée. Même principe pour les deux autres critères : la disponibilité peut être présentée par les valeurs linguistiques suivantes : faible, moyenne et

élevée et le critère de temps de réponse peut être présenté par les valeurs linguistiques : petit, moyen et long.

Notons que cette étape est principalement réalisée sur la base d'observations statistiques (ou par apprentissage pour regrouper les valeurs d'une variable en catégories homogènes) ou sur l'avis des experts.

En bref, cette première étape consiste à transformer les variables (d'entrée et de sortie) en variables linguistiques :

— Pour chaque variable, on définit dans un premier temps l'univers du discours (i.e. la plage de valeurs que peut prendre la variable) ;

— La variable est ensuite découpée en catégories appelées valeurs linguistiques ;

— Une fonction (allant de 0% à 100%) permettant de définir pour chaque valeur son pourcentage de véracité à l'affirmation : « l'observation est dans telle catégorie » est affectée à chaque catégorie.

Puisque nous avons choisi la fonction d'appartenance triangulaire pour présenter nos données linguistiques, le tableau suivant est un exemple qui montre la présentation des critères fiablilité, Disponibilité et temps de réponse en valeurs linguistiques :

Terme linguistique	Nombre triangulaire flou
Reliability	
low	(0,0,3)
medium	(2,5,8)
high	(6,9,10)
Availability	
low	(0,0,4)
medium	(2,5,7)
high	(5,8,10)
Response time	
small	(0,0,4)
medium	(2,5,8)
long	(6,10,10)

TABLE 6.1 – Critères en terme linguistique

Les fonctions d'appartenance triangulaires de chaque variables est représenté comme suit :

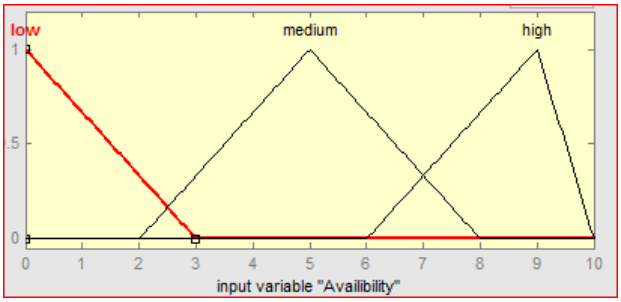

FIGURE 6.4 – Variable availability

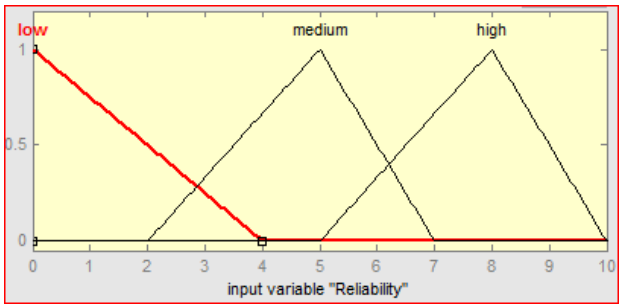

FIGURE 6.5 – Variable reliability

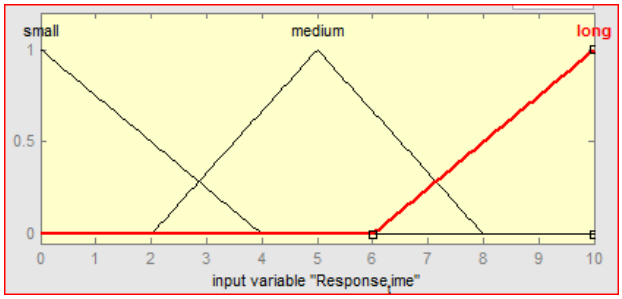

FIGURE 6.6 – Variable response-time

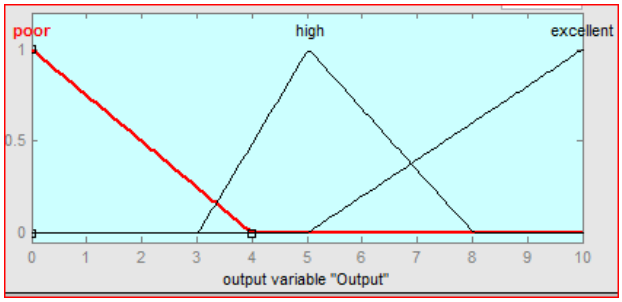

FIGURE 6.7 – Variable Output

Moteur d'inférence flou

L'entrée du moteur d'inférence floue est une matrice de décision floue résultant de la fuzzification dont les valeurs linguistiques correspondent à chaque service Web.

$$\widetilde{M} = \begin{array}{c} \\ s_1 \\ s_2 \\ .. \\ s_m \end{array} \begin{array}{c} \begin{array}{cccc} c_1 & c_2 & ... & c_n \end{array} \\ \begin{pmatrix} \widetilde{x}_{11} & \widetilde{x}_{12} & ... & \widetilde{x}_{1n} \\ \widetilde{x}_{21} & \widetilde{x}_{22} & ... & \widetilde{x}_{2n} \\ ... & ... & ... & ... \\ \widetilde{x}_{m1} & \widetilde{x}_{m2} & ... & \widetilde{x}_{mn} \end{pmatrix} \end{array}$$

Où s_1, s_2, ..., s_m sont les services Web, c_1, c_2, ..., c_n sont les paramètres de qualité de service et x_{ij} est une valeur floue présentant la valeur de qualité c_j pour le service s_i.

Pour chaque service Web, ses données sont transmises au moteur d'inférence flou qui applique l'ensemble des règles floues pour son raisonnement. Chaque règle est décrite en fonction des connaissances qu'elle possède. Pour son fonctionnement, il applique chaque règle aux variables linguistiques calculées dans l'étape Fuzzification et le résultat de cette étape est une valeur floue dite score flou caractérisant chaque service Web.

La base de règles floues

Ces règles ont la forme (si X alors Y), elles sont créées par le concepteur et seront exploitées par le moteur d'inférence flou pour produire un résultat flou qui sera traduit plus tard pour donner une valeur quantitative. Les opérateurs les plus utilisés dans l'évaluation des règles sont l'union qui est traduite par MAX et l'intersection qui est traduite par MIN.

Exemple de règles flou

1. If (Availability is low) and (Reliability is low) then (output1 is poor)
2. If (Availability is high) and (Reliability is high) then (output1 is excellent)
3. If (Availability is high) then (output1 is high)
4. If (Reliability is high) then (output1 is high)
5. If (Availability is medium) and (Reliability is low) then (output1 is poor)
6. If (Availability is medium) and (Reliability is high) then (output1 is high)
7. If (Availability is high) and (Reliability is medium) then (output1 is high)
8. If (Availability is low) and (Reliability is high) then (output1 is poor)
9. If (Availability is low) then (output1 is poor)

10. If (Reliability is low) then (output1 is poor)

Pour chaque règle, le moteur d'inférence va attribuer un coefficient de modification d'un sous-ensemble, mais il peut très bien donner, pour des règles différentes, des coefficients de modification différents pour un même sous-ensemble de sortie. Dans ce cas pour trouver la valeur du coefficient de modification de sortie du sous-ensemble considéré, le moteur d'inférence va effectuer un OU entre ces différents coefficients de modification.

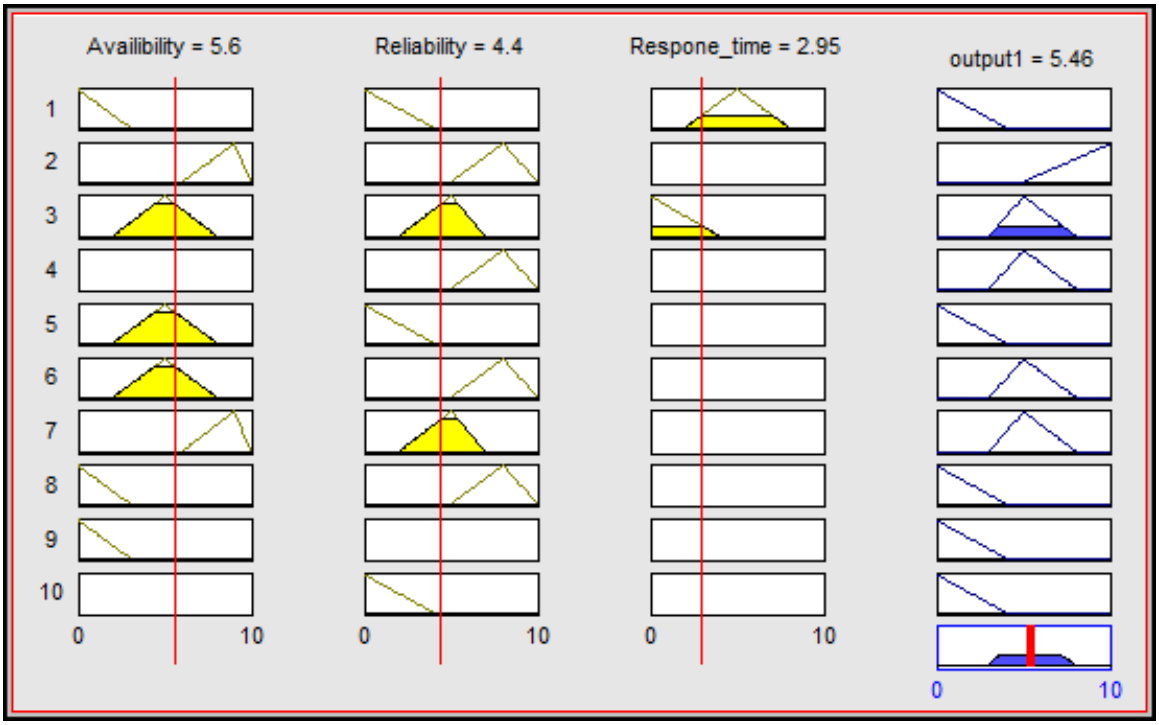

FIGURE 6.8 – Application des règles

Defuzzification

Contrairement à la fuzzification, la défuzzification permet d'associer à chaque valeur floue, qui correspond à la sortie désirée, une valeur réelle et concrète. Cette étape peut être effectuée de plusieurs façons selon le concept mathématique choisi. Nous signalons qu'il existe plusieurs méthodes pour la déffuzification :

— Center of Sums Method (COS)

— Center of gravity (COG)

— Centroid of Area (COA) method

— Bisector of Area Method (BOA)
— Weighted Average Method
— Mean of Maxima Method (MOM)

6.3 Une approche basée agent et logique floue pour la sélection des services Web

Cette approche est axée principalement sur un algorithme de prise de décision multi-critères flou(MCDM flou).

6.3.1 Méthode de prise de décision multi-critères floue (MCDM)

Le problème de la prise de décision est le processus consistant à trouver la meilleure option parmi toutes les alternatives possibles[98]. Le problème typique du MCDM est de traiter l'évaluation d'un ensemble d'alternatives en fonction d'un ensemble de critères de décision[77].

L'utilisation de ce type de solution est due aux points suivants[10] :

— L'évaluation basée sur un seul critère ne couvre pas toutes les facettes du problème.
— La règle de la somme pondérée présente des problèmes de compensation et de changement d'échelle.
— La plupart des critères de qualité de service pour les services Web sont qualitatifs et la somme pondérée n'est pas pertinente pour ce type de critère.

La prise de décision multicritère (MCDM) est prise en charge pour optimiser la sélection de service en utilisant des contraintes de QoS, en fusionnant plusieurs propriétés de ressources[78].

Plusieurs algorithmes MCDM existent dans la littérature : modèle de la somme pondérée, modèle du produit pondéré, ELECTRE[103], AHP[104] et TOPSIS[77].

6.3.2 Approche proposée

Initialement, le client initie sa requête qui spécifie ses besoins exprimés en langage naturel. L'agent client analyse ce dernier et le normalise sous la forme d'une contrainte fonctionnelle. L'agent Enregistrement renverra une liste contenant les services Web répondant à la requête

émise. L'agent QoS Manager extrait pour chaque service candidat les valeurs quantitatives des paramètres non fonctionnels tels que la disponibilité, la fiabilité, le temps de réponse et le coût. Ensuite, pour chaque service candidat, ses paramètres sont transmis au sous-système flou.

Ce sous-système comprend principalement deux types d'agents : un agent (agent de fuzzification $_1$, agent de fuzzification $_2$, ..., agent de fuzzification $_n$) et un agent de classement flou.

Chaque agent de fuzzification i est responsable de la transformation du critère quantitatif i en valeurs linguistiques en utilisant une fonction d'appartenance donnée. Pour notre approche, nous utilisons la présentation de nombres triangulaires flous.

L'utilisation de cette présentation est justifiée par le fait que la traduction de l'expertise humaine vers ce type de nombre flou est plus facile.

Nous obtenons alors une matrice floue ; cette matrice a comme lignes les services Web et en colonne les critères de qualité et l'intersection de la ligne i et la colonne j donnera la valeur floue x_{ij} qui représente la valeur de la qualité j pour le service i.

Par la suite, ce dernier sera mis à la disposition de l'agent classement flou , qui utilise l'algorithme Fuzzy TOPSIS pour classer la liste des services Web en fonction des critères afin de fournir le meilleur service qui sera envoyé à l'agent Client qui est ensuite mis à disposition au client final pour puisse invoquer le service.

La figure suivante montre l'architecture de l'approche proposée.

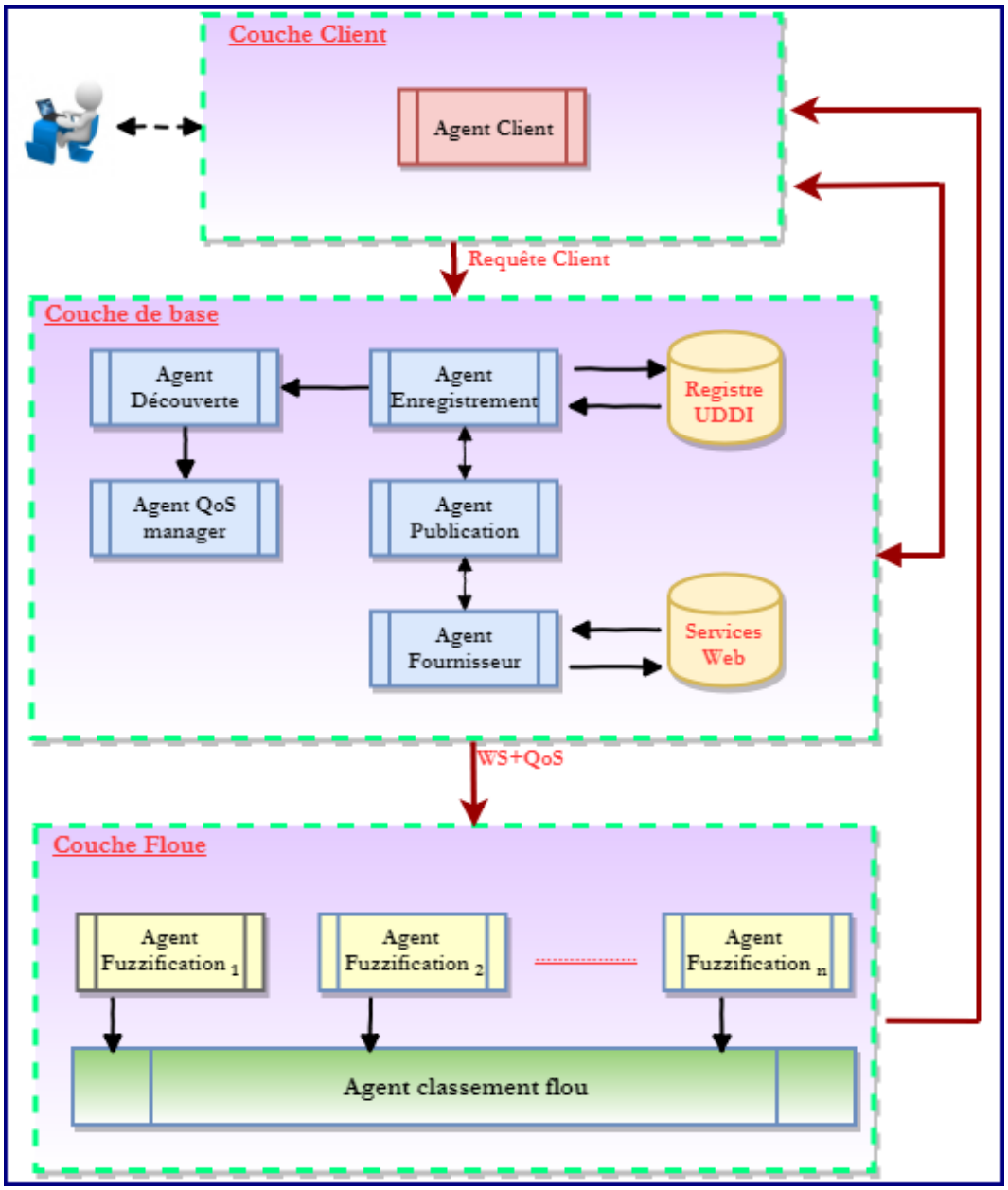

FIGURE 6.9 – Architecture de l'approche proposée

6.3.3 Description des différents agents

Dans cette sous section, nous allons présenter les architectures de chaque agent.

Agent Client

C'est l'agent qui permet à l'utilisateur d'interagir avec le système. Il agit comme un intermédiaire entre l'utilisateur et les autres agents (Agent Enregistrement et Agent Fourniseur), vérifie la requête émise par l'utilisateur, affiche le résultat final à l'utilisateur. Il demande un service. Il saisit les exigences fonctionnelles des services souhaités tels que vacances, météo, email ... etc.

L'architecture de l'agent client est la suivante :

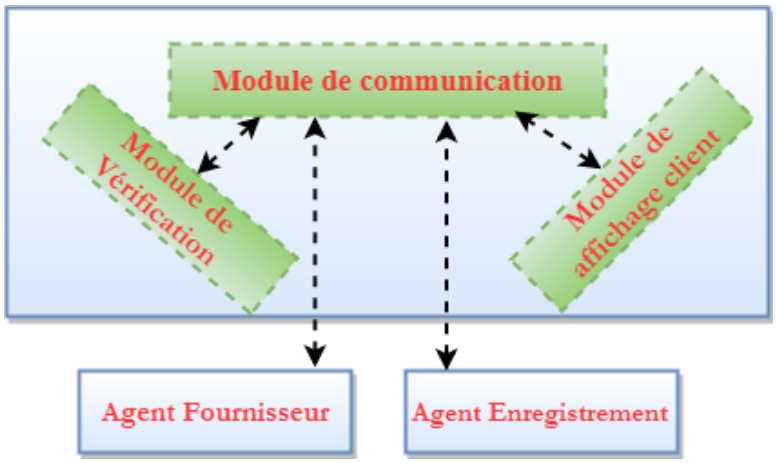

FIGURE 6.10 – Architecture de l'agent Client

Module de communication : Ce module est responsable de l'échange de messages entre l'agent client et les autres agents.

Module de Vérification : Ce module est responsable de la demande de l'utilisateur, de sa vérification ou de l'adaptation au format général à envoyer à l'agent enregistrement.

Module affichage Client : Ce module est un intermédiaire entre le client et le système. Il permet au client d'entrer sa demande et de l'envoyer à l'agent adéquat via le module de communication et d'afficher les résultats des autres agents.

Agent découverte

C'est un agent intermédiaire qui, après que l'agent client a lancé sa demande, l'agent Découverte reçoit une liste de services similaires du registre UDDI et répond à cette requête.

L'architecture l'agent découverte est la suivante :

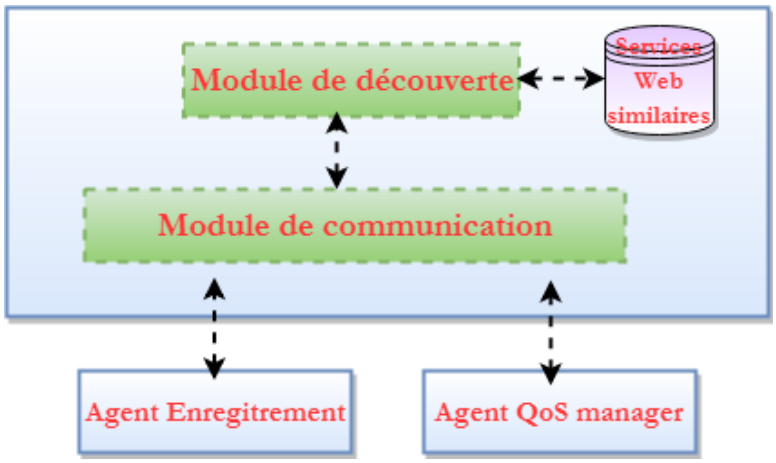

FIGURE 6.11 – Architecture de l'agent Découverte

Module découverte : Ce module récupère le résultat de la requête émise par l'agent client en le plaçant dans une base de données.

Agent QoS Manager

L'agent QoS Manager est un agent qui permet d'extraire pour chaque Service Web les valeurs de ses critères de qualité.

L'architecture de l'agent QoS Manager est la suivante :

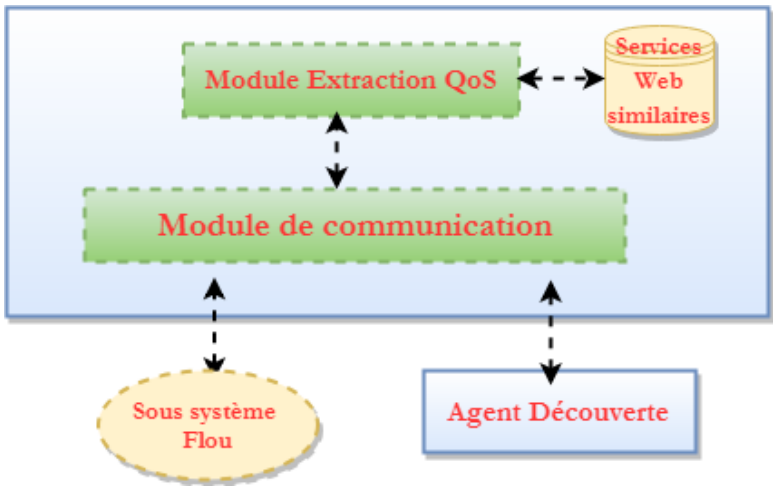

FIGURE 6.12 – Architecture de l'agent QoS Manager

Module Extraction QoS : Ce module se charge pour chaque service Web d'extraire les valeurs de critères qui les constituent à envoyer au sous-système flou.

Agent Eregistrement

L'agent Enregistrement est conçu pour héberger des informations sur les services Web de manière structurée dans l'annuaire des registres. Grâce à cet agent, il devenait possible de publier et de découvrir des informations sur une entreprise et ses services Web. Il contient des informations sur les services telles que le nom du service, l'URL et les opérations.

L'architecture de l'agent Enregistrement est la suivante :

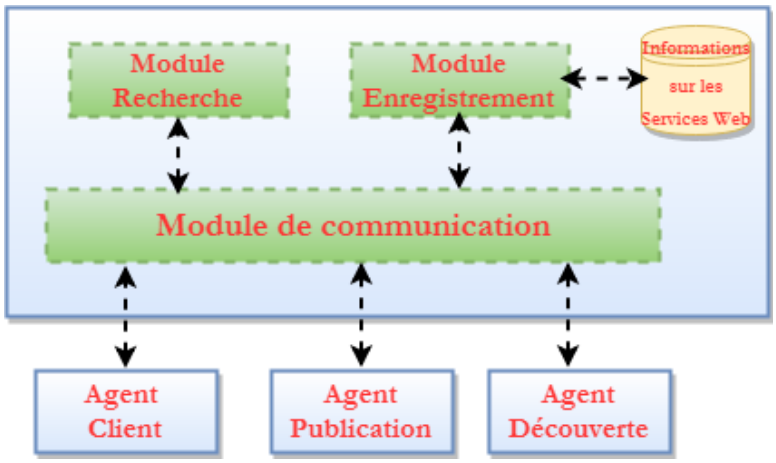

FIGURE 6.13 – Architecture de l'agent Enregistrement

Module de Recherche : Ce module a pour fonction de traiter la requête envoyée par l'agent Client.

Module Enregistrement : Ce module est utilisé pour publier des services Web dans le registre approprié afin de faciliter sa découverte.

Agent Publication

Cet agent publie les informations relatives aux services Web y compris les informations de QoS provenant de l'agent fournisseu auprès de l'agent Enregistrement après la certification de la QoS.

L'architecture de l'agent Publication est la suivante :

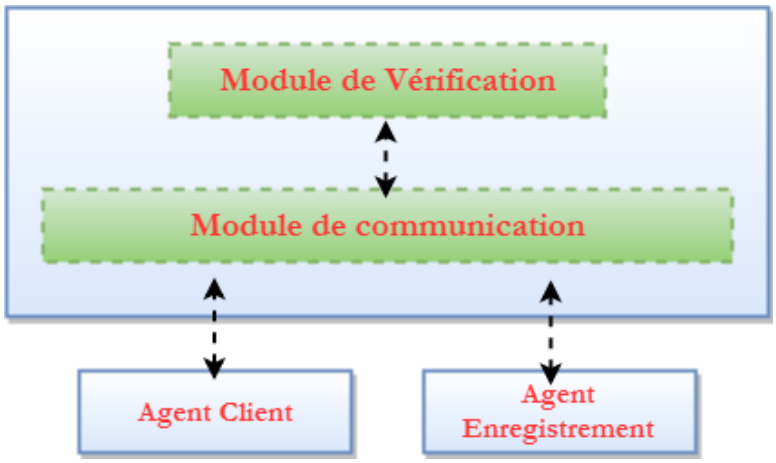

FIGURE 6.14 – Architecture de l'agent Publication

Agent fournisseur

Cet agent implémente les services et les publie à travers l'agent Publication. Il entre les informations décrivant les services, y compris les différents critères non fonctionnels.

L'architecture de l'agent fournisseur est la suivante :

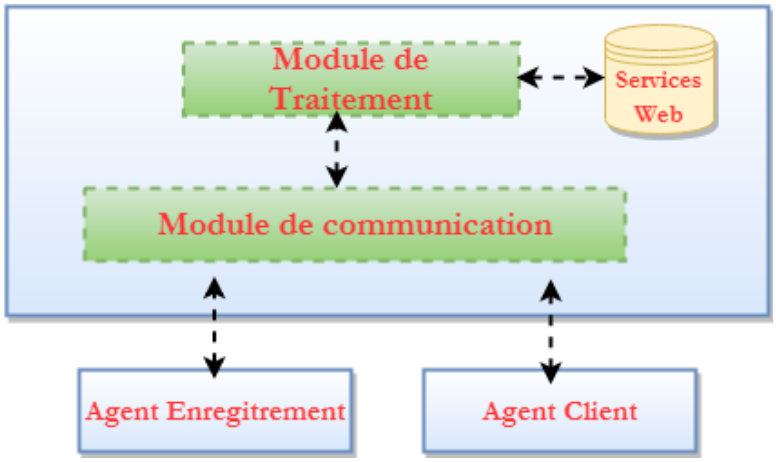

FIGURE 6.15 – Architecture de l'agent Fournisseur

La couche Floue

Dans cette couche, le nombre d'agents de fuzzification est similaire au nombre de critères qualifiant les services Web dont le rôle de chaque agent est de fuzzifier le critère i. le fait de

mettre plusieurs agents de fuzzification est de travailler dans le parallélisme et par conséquent de minimiser le temps de fuzzification notamment lorsqu'il s'agit de plusieurs critères.

Le sous-système décrivant cette couche peut être représenté comme suit :

Il a en entrée les services Web avec leurs critères et en sortie, le résultat de classement qui correspond au meilleur service. Il consiste en un ensemble d'agents effectuant la même fonction (Fuzzification) et un autre agent chargé d'appliquer l'algorithme de classement. Le fonctionnement de cette couche peut être représenté comme suit :

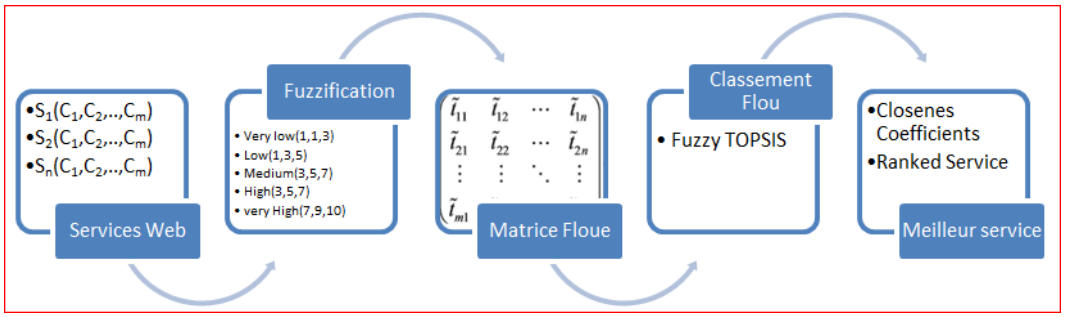

FIGURE 6.16 – Fonctionnement de la couche floue

Agent Fuzzification :

Le but de cet agent est de transformer les résultats quantitatifs obtenus à partir de de l'agent QoS Manager en variables linguistiques.

Pour ce faire, le concepteur de système flou doit créer des fonctions d'appartenance. Une fonction d'appartenance est une fonction qui permet de définir le degré d'appartenance d'une donnée numérique à une variable linguistique. Le résultat global de la fuzzification est une matrice de décision floue qui sert comme entrée floue au sous-module classement flou.

L'architecture de l'agent de Fuzzification agent est la suivante :

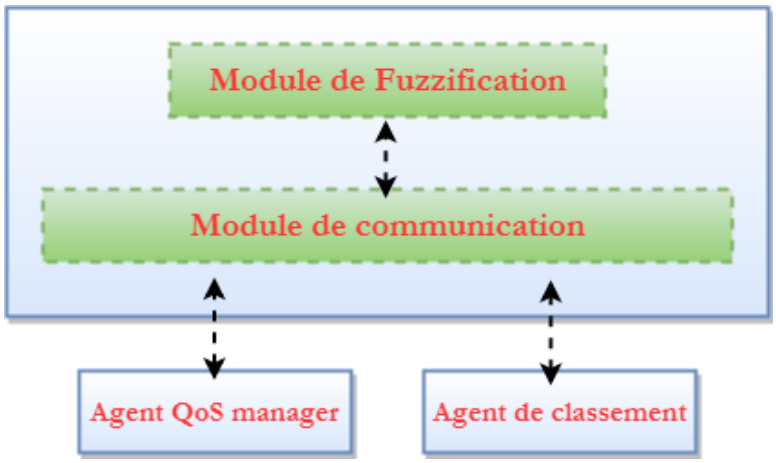

FIGURE 6.17 – Architecture de l'agent fuzzification

L'évaluation des critères et des alternatives (services Web) est représentée dans des valeurs linguistiques telles que : bon, mauvais, excellent, très pauvre, pauvre, passable, bonne, très bonne et sont expliquées dans des nombres triangulaires flous.

Agent classement flou :

Une fois, il reçoit toutes les valeurs linguistiques des différents critères provenant des différents agents de fuzzification. Pour chercher le meilleur service Web à rendre au client, il se focalise sur l'algorithme Fuzzy TOPSIS. Il a comme entrée la matrice de décision floue formée des services Web et de leur qualité de services et produit le coefficient de proximité(score) de chaque Web.

L'architecture de l'agent classemnt flou est la suivante :

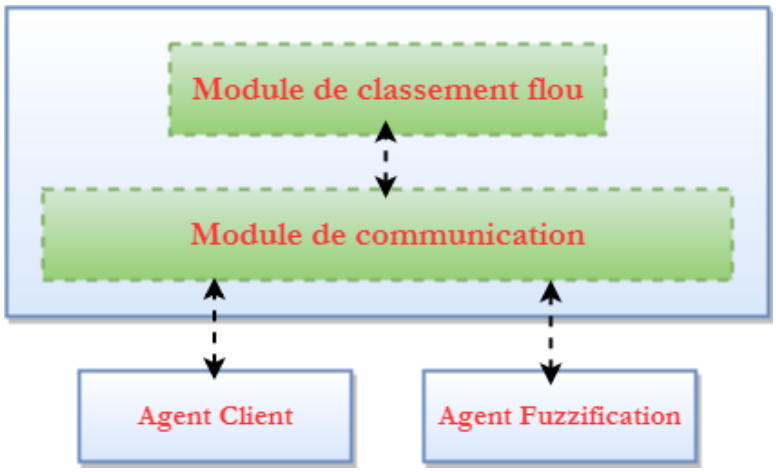

FIGURE 6.18 – Architecture de l'agent Ranking

Matrice de decision Floue

Nous considérons un ensemble de services Web (S_1, S_2, .., S_n) et un ensemble de critères (C_1, C_2, .., C_m), tous les services sont caractérisés par les mêmes critères.

Dans ce système, nous avons les services Web disponibles avec leurs qualités de service exprimées en termes flous; nous prenons la présentation triangulaire floue pour les fonctions d'appartenance.

Les poids de différents critères et la notation des critères peuvent être exprimés en termes linguistiques.

x_{ij} est l'indice de performance du service S_i (i = 1,2, ... n) sous le critère C_j (j = 1,2 .., m). x_{ij} est exprimé sous la forme floue comme suit (a_{ij}, b_{ij}, c_{ij}) Lorsque a_{ij} représente la valeur inférieure,b_{ij} représente la valeur modale et c_{ij} représente la valeur supérieure de l'attribut.

Nous obtenons la matrice de décision suivante : $\widetilde{M} =$

$$\begin{array}{c} \\ s_1 \\ s_2 \\ .. \\ s_m \end{array} \begin{array}{c} \begin{array}{cccc} c_1 & c_2 & ... & c_n \end{array} \\ \begin{pmatrix} \widetilde{x}_{11} & \widetilde{x}_{12} & ... & \widetilde{x}_{1n} \\ \widetilde{x}_{21} & \widetilde{x}_{22} & ... & \widetilde{x}_{2n} \\ ... & ... & ... & ... \\ \widetilde{x}_{m1} & \widetilde{x}_{m2} & ... & \widetilde{x}_{mn} \end{pmatrix} \end{array}$$

W_j est le poids du critère c_j, il est exprimé en logique floue comme suit $w_j = (y_{ij}, z_{ij}, t_{ij})$

L'approche TOPSIS est actuellement l'une des méthodes les plus populaires (MCDM) et s'est avérée fournir des résultats utiles dans diverses applications.

6.3.4 Algorithme TOPSIS flou

L'algorithme TOPSIS floue est utilisé dans plusieurs domaines d'application : problème de localisation, sélection des fournisseurs et énergie durable et renouvelable[79].

Son principe est de choisir une alternative plus proche de la solution idéale et la plus éloignée de la solution négative idéale. La solution positive idéale est composée des meilleures valeurs de performance pour chaque critère, tandis que la solution négative idéale consiste en les plus mauvaises performances.

Le classement des alternatives dans TOPSIS est basé sur «la similarité relative à la solution idéale», ce qui évite d'avoir la même ressemblance avec les idéal positif et l'idéal négatif[80][81]. L'algorithme TOPSIS flou comprend les étapes suivantes :

1. Attribution des évaluations aux critères et aux services et calculer leur évaluations floues agrégées.

   Nous supposons que nous avons n services Web $(s_1, s_2, .., s_n)$ qui sont évalués selon les critères $C_j (j = 1, 2, .., m)$.

   L'évaluation du service Web S_i selon le critère C_j est notée par : $\widetilde{X}_{ij} = (a_{ij}, b_{ij}, c_{ij})$ et le poids du critère C_j est noté $W_{ij} = (w_{j1}, w_{j2}, w_{j3})$.

2. Cacluler la matrice de décision floue normalisée. la matrice normalisée floue $\widetilde{R}$ est donnée par :

   Pour les critères positifs : $\widetilde{r}_{ij} = (\frac{a_{ij}}{c_j^*}, \frac{b_{ij}}{c_j^*}, \frac{c_{ij}}{c_j^*})$ and $c_j^* = \max_i \{c_{ij}\}$

   Pour les critères négatifs : $\widetilde{r}_{ij} = (\frac{a_j^-}{c_{ij}}, \frac{a_j^-}{b_{ij}}, \frac{a_j^-}{c_{ij}})$ et $c_j^* = \min_i \{a_{ij}\}$

3. Calculer la matrice normalisée pondérée

   $\widetilde{V} = [\widetilde{v}_{ij}]_{mxn}$ where $\widetilde{v}_{ij} = \widetilde{r}_{ij}.\widetilde{w}_{ij}$

4. Calculer la solution Idéale Positive Floue (FPIS) et la solution Idéale Négative Floue négative (FNIS) :

   Le FPIS est calculé comme suit : $A^* = (\widetilde{v}_1^*, \widetilde{v}_2^*, ..., \widetilde{v}_3^*)$ où $\widetilde{v}_j^* = \max_i \{v_{ij3}\}$

   le FNIS est calculé comme suit : $A^- = (\widetilde{v}_1^-, \widetilde{v}_2^-, ..., \widetilde{v}_3^-)$ où $\widetilde{v}_j^- = \min_i \{v_{ij1}\}$

5. Calculer la mesure de séparation pour FPIS et FNIS : :

   $d_i^* = \sum_{j=1}^{n} d(\widetilde{v}_{ij}, \widetilde{v}_j^*)$ où $i = 1, 2, 3, ..., m$

$d_i^- = \sum_{j=1}^{n} d(\widetilde{v}_{ij}, \widetilde{v}_j^-)$ où $i = 1,2,3,...,m$

$d(\widetilde{x}, \widetilde{y})$ est la mesure de distance entre des nombres flous $\widetilde{x}$ and $\widetilde{y}$.

6. Calculer le coefficient de proximité pour chaque service S_i ; ce coefficient représente les distances à la solution idéale positive floue S^* et la solution idéale négative floue S^- :

   $CC_i = \frac{d_i^*}{d_i^- + d_i^*}$

7. Classer les services.

6.4 Conclusion

Dans ce chapitre, nous avons présenté les approches proposées pour résoudre la problématique liée à la prise en charge des données linguistiques des utilisateurs lors de la découverte des services Web.

Dans la première proposition, nous avons présenté un modèle qui s'appuit sur la théorie de base d'un système floue. il est constitué principalement d'un sous-système floue composé de trois modules : Fuzzification, moteur d'inférence et deffuzification et plus d'une base de règles floues formée de règles dictées en langage naturel. Pour chaque service Web, ce système accordera un score qui sert comme qualifiant.

Par contre, dans la deuxième proposition, nous avons présenté une approche qui s'appuit sur la logique floue et le paradigme agent. Elle est comopsée de trois couche : couche client, couche de base et couche floue. Cette dernière se compose de deux agents qui colaborent ensemble en utilisant un algorithme de décision multi critères pour classer les services Web candidats.

Dans le chapitre et afin de montrer la faisabilité des ces approches, nous allons montrer une experimentation.

Chapitre 7

Résultats Expérimentaux et Discussions

7.1 Introduction

Dans cette section, nous proposons de réaliser deux exemples d'illustration. Le premier exemple concerne la première propposition décrite dans le chapitre contribution tandis que le deuxième exemple d'illustartion concerne la deuxième proposition.

7.2 Exemple 1 :(Un modèle de sélection des services Web basée sur la logique floue)

Dans cet exemple, nous proposons une expérimentation dont la mise en œuvre a été réalisée avec MATLAB pour démontrer la faisabilité de notre proposition. Ces données (services Web et leurs qualités) sont basées sur le test de[94].

L'exemple est décrit comme suit (tableau 7.1) : Nous avons six services Web (ABC, BTC, WS_1, WS_2, WS_3 et WS_4). Ces services Web sont examinés selon 7 critères : Prix (Pe), disponibilité (Av), délai d'attente (To), taux de rémunération (Cr), taux de pénalité (Pr), durée d'exécution (Ed), réputation (Re).

	Pe	Av	To	Ed	Re	Cr	Pr
ABT	25	0.7	75	100	3	0.5	0.5
BTC	40	0.8	200	40	2.5	0.8	0.1
WS_1	46	0	65	60	1	0.7	0.4
WS_2	38	0.8	120	25	4	0.85	0.3
WS_3	27	0.9	95	30	3	0	0.1
WS_4	30	0.75	180	85	3	0.95	0.2

TABLE 7.1 – Informations des QoS dans le service fournisseur (Exemple 1)

La requête émise par l'utilisateur est donnée par : (Price ≤ 50, availability≥ 0.75, timeout≥ 70, Compensation rate≥ 0.7, Penalty rate≤0.45, Execution duration≤ 100, reputation≥ 2.5).

Les deux services Web ABT et WS_1 sont éliminés de la sélection car ils ne répondent pas aux exigences de l'utilisateur.

La première étape de notre proposition consiste à présenter chaque variable numérique (critère de qualité) dans des valeurs floues qui sont essentielles pour effectuer la fuzzification.

Dans le tableau suivant nous présenterons les termes linguistiques correspondant aux fonctions d'appartenance de chaque variable. Notons ici que nous avons choisi le modèle triangulaire flou pour présenter nos données.

Terme linguistique	Nombre flou triangulaire
Price	
cheap	[0 15 30]
acceptable	[20 40 60]
expensive	[50 70 100]
Availability	
low	[0 0 0.3]
medium	[0.2 0.35 0.5]
high	[0.4 0.6 0.8]
excellent	[0.7 0.9 1]
Time Out	
low	[0 50 80]
medium	[50 120 170]
great	[130 200 250]
Execution duration	
small	[0 0 80]
medium	[50 100 160]
long	[120 160 200]
Reputation	
bad	[0 0 2]
good	[1.5 2.5 3.8]
very good	[2.8 4 5]
Compensation rate	
bad	[0 0 0.3]
good	[0.1 0.5 0.9]
very good	[0.6 1 1]
Penalty rate	
bad	[0 0 0.3]
good	[0.1 0.4 0.7]
very good	[0.5 0.8 1]

TABLE 7.2 – Critères linguistiques

Ces variables sont présentées linguistiquement par la méthode nombre flou triangulaire comme suit :

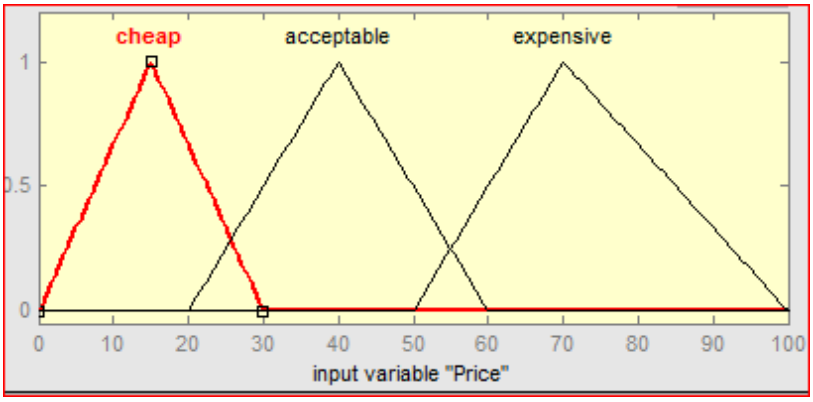

FIGURE 7.1 – Variable linguistique Price

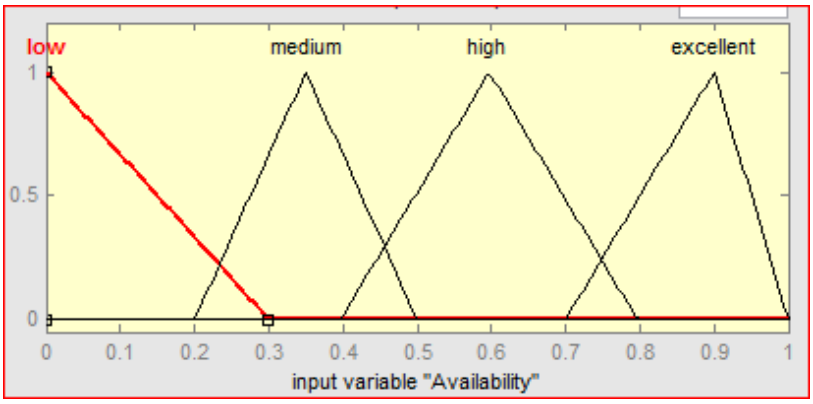

FIGURE 7.2 – Variable linguistique Availability

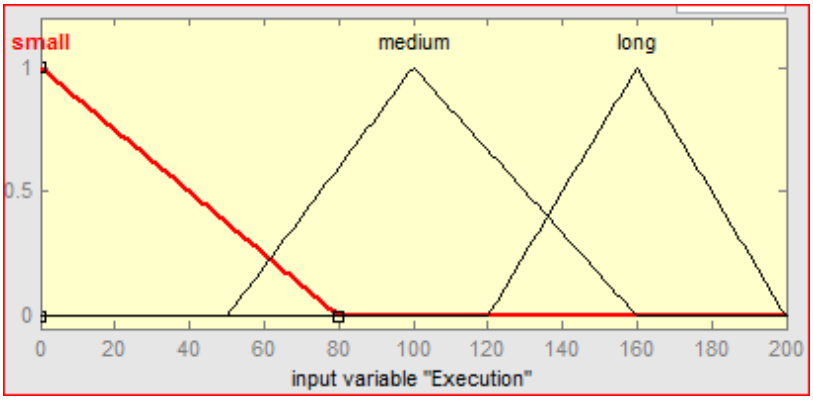

FIGURE 7.3 – Variable linguistique Execution duration

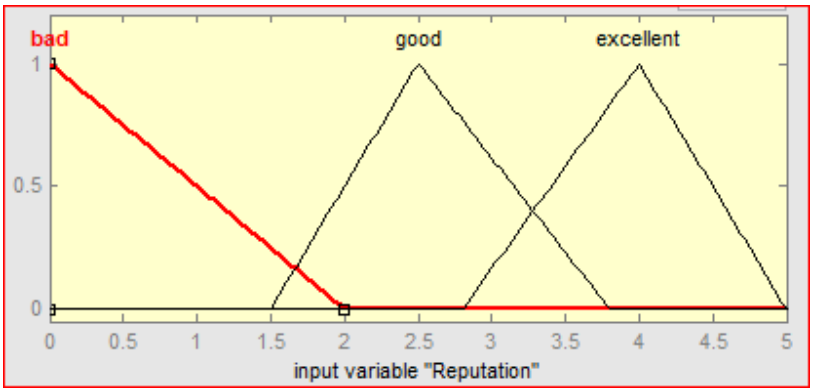

FIGURE 7.4 – Variable linguistique Reputation

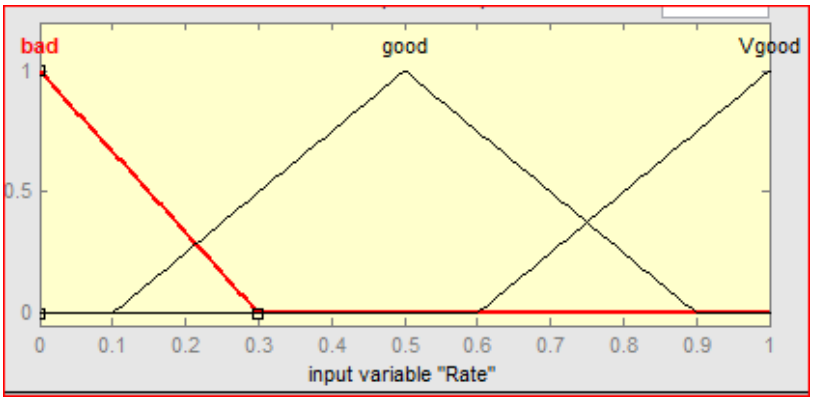

FIGURE 7.5 – Variable linguistique Compensation rate

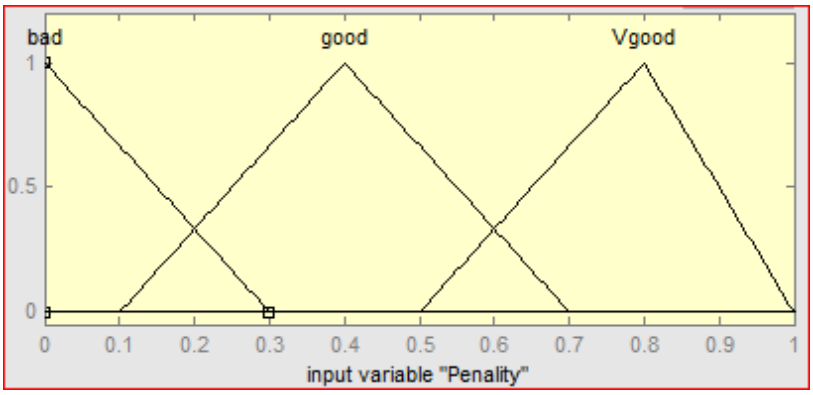

FIGURE 7.6 – Variable linguistique Penality Rate

Enfin, la variable linguistique qualifiant le score relatif à chaque service Web est défini comme suit :

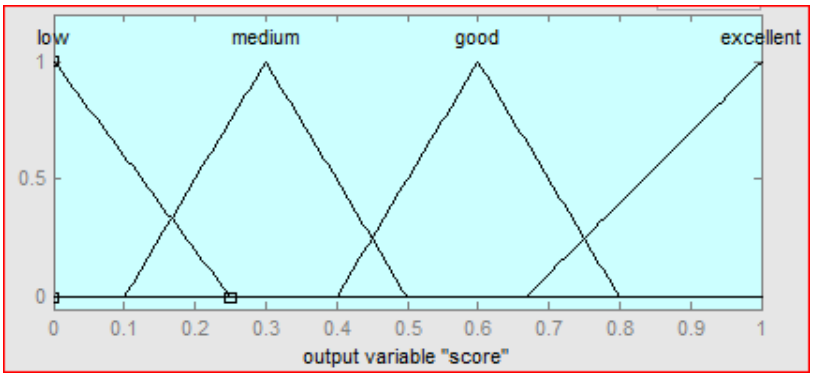

FIGURE 7.7 – Variable score

En utilisant la logique floue, les besoins des utilisateurs peuvent être représentés comme suit : :(Price=cheap, availability=excellent, timeout=low or medium, Compensation rate=very good, Penalty rate= cheap or acceptable, Execution duration= small or medium, reputation= good or excellent).

Dans cette exemple d'illustration, nous avons écrit un ensemble de règles qui seront appliquées par le moteur d'inférence floue. Dans la suite, nous présentons quelques règles :

1. If (Availability is low) then (score is low).
2. If (Price is cheap) and (Availability is excellent) and (TimeO is low) and (Execution is small) and (Reputation is good) and (Rate is good) and (Penality is bad) then (score is excellent).
3. If (Price is acceptable) and (Availability is medium) and (TimeO is medium) and (Execution is medium) and (Reputation is good) and (Rate is good) and (Penality is good) then (score is medium).
4. If (Price is expensive) and (Availability is low) and (TimeO is great) and (Execution is long) and (Reputation is bad) and (Rate is good) and (Penality is bad) then (score is low).
5. If (Price is cheap) and (Availability is excellent) and (TimeO is low) and (Execution is small) and (Reputation is excellent) and (Rate is bad) and (Penality is bad) then (score is excellent).
6. If (Price is acceptable) and (Availability is high) and (TimeO is medium) and (Execution is medium) and (Reputation is excellent) and (Rate is good) and (Penality is good) then (score is good).

7. If (Price is cheap) and (Availability is excellent) and (TimeO is great) and (Execution is long) and (Reputation is excellent) and (Rate is good) and (Penality is good) then (score is medium).
8. If (Price is expensive) and (Availability is low) and (TimeO is great) and (Execution is long) and (Reputation is bad) and (Rate is not good) and (Penality is not good) then (score is low).
9. If (Price is acceptable) and (Availability is high) and (TimeO is medium) and (Execution is medium) and (Reputation is excellent) and (Rate is Vgood) and (Penality is bad) then (score is good).
10. If (Price is cheap) and (Availability is excellent) and (TimeO is medium) and (Execution is long) and (Reputation is excellent) and (Rate is Vgood) and (Penality is Vgood) then (score is excellent).
11. If (Price is cheap) and (Availability is excellent) and (TimeO is medium) and (Execution is small) and (Reputation is excellent) and (Rate is bad) and (Penality is bad) then (score is excellent).
12. If (Price is acceptable) and (Availability is excellent) and (TimeO is medium) and (Execution is small) and (Reputation is excellent) and (Rate is good) and (Penality is good) then (score is good).

En exécutant le moteur d'inférence qui applique toutes les règles pour chaque service Web, nous obtenons les scores en valeurs quantitatives de chaque service Web :

	Pe	Av	To	Ed	Re	Cr	Pr	score
BTC	40	0.8	200	40	2.5	0.8	0.1	**0.5**
WS_2	38	0.8	120	25	4	0.85	0.3	**0.6**
WS_3	27	0.9	95	30	3	0	0.1	**0.849**
WS_4	30	0.75	180	85	3	0.95	0.2	**0.5**

TABLE 7.3 – Scores attribués à chaque service Web

Comme, nous voyons la dernière colonne du tableau précédent, le meilleur service est celui qui a le plus de score donc le service WS_3 est le service classé alors il sera retourné à l'utilisateur.

Comparaison avec les méthodes conventionnelles

Nous avons proposé un algorithme de sélection de service Web basé sur la logique floue. Le classement a été principalement effectué par un moteur d'inférence flou qui utilise en entrée une matrice floue constituée de services Web similaires avec leurs paramètres de qualité correspondants et avec la base des règles floue qu'il contient, il s'applique dans ce cas à tous les services Web, chacun un à part pour donner à la fin, un score caractérisant chaque entrée (WS + QoS) qui sera utilisé pour le classement et effectivement renvoyé à l'utilisateur.

Le modèle a été évalué en classant un ensemble de services Web similaires à l'aide de sept critères de qualité de service décrits dans les valeurs linguistiques. Ils sont classés par l'utilisation du moteur flou (tableau 7.3).

Si nous comparons les résultats obtenus avec les méthodes conventionnelles[94], nous pouvons dire que les résultats données par les deux méthodes sont proches (tableau 7.4). Ces résultats peuvent être améliorés si nous utilisons plus de règles. Encore, ces résultats seront plus performants si les règles on été dictées par des experts de domaines.

Service Web	Classement flou	Classement Conventionnel
BTC	0,4458	0,5
WS_2	0,5713	0,6
WS_3	0,8457	0,849
WS_4	0,568	0,5

TABLE 7.4 – Classement flou et Classement conventionnel

Ce résultat peut être représenté par un graphique comme suit :

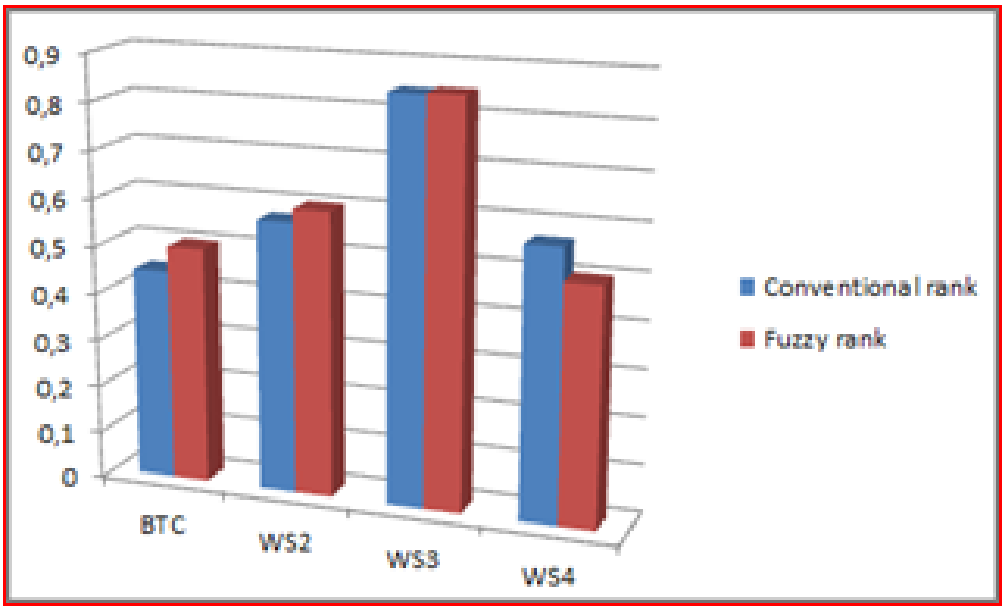

FIGURE 7.8 – Classement des services

Les résultats obtenus dans le tableau ci-dessus sont illustrés graphiquement sur la figure 7.8 et à partir de cette figure, nous déduisons également les mêmes résultats. Par rapport aux méthodes conventionnelles, la méthode floue aide facilement les clients à sélectionner des services Web tout en proposant une requête en termes linguistiques. Ceci encourage notamment les utilisateurs non qualifiés des paramètres de critères de qualité à choisir les meilleurs services en termes de qualité. Nous pouvons ajouter aussi que le point fort de cette solution demeure dans l'utilisation des règles en langage naturel.

7.3 Exemple 2 :(Une approche basée agent et logique floue pour la sélection des services Web)

Nous utilisons cet exemple pour démontrer la faisabilité de notre deuxième proposition. L'exemple est décrit comme suit (Tableau 7.5) : Nous disposons de cinq services Web : (WS_0, WS_1, WS_2, WS_3 et WS_4) avec la même fonctionnalité. Ces services Web sont examinés selon 4 critères : fiabilité (rel), disponibilité (Av), prix (pr) et temps de réponse (rt) dont les deux premiers critères fiabilité et de disponibilité sont favorables et les deux qui restent sont défavorables.

	WS_0	WS_1	WS_2	WS_3	WS_4
Rel	90,01	41,38	61,47	29,55	45,12
Av	83,81	87,2	86,73	95,31	12,81
pr	5,24	3,89	14,67	6,24	6,19
Rt	715,58	1453,33	140,27	988,2	629,9

TABLE 7.5 – Informations des QoS dans le fournisseur (exemple 2)

Application de l'algorithme TOPSIS floue

Par application de l'algorithme TOPSIS floue décrit dans le chapitre précédent :

Fuzzification des critères de services :

En utilisant la méthode triangulaire des nombres flous, nous pouvons obtenir la présentation linguistique de chaque paramètre :

terme linguistique	**nombre triangulaire flou**
Cheap	[2 2 8]
Medium	[6 10.5 15]
Expensive	[12.8 16 20]

TABLE 7.6 – Paramètre Price en présentation Floue

terme linguistique	**nombre flou triangulaire**
Low	[20 20 450]
Medium	[300 600 900]
High	[700 900 1200]
Very high	[1000 1300 1500]

TABLE 7.7 – Paramètre Response time en présentation Floue

terme linguistique	**nombre flou triangulaire**
Low	[0 0 25]
Fair	[15 30 50]
Good	[35 50 70]
Very good	[60 80 100]

TABLE 7.8 – Paramètre Availability en présentation Floue

terme linguistique	**nombre flou triangulaire**
Low	[0 0 25]
Fair	[15 30 50]
Good	[35 50 70]
Very good	[60 80 100]

TABLE 7.9 – Paramètre Reliability en présentation Floue

La matrice de décision floue est donnée par :

$$\widetilde{M} = \begin{array}{c} \\ rel \\ Av \\ pr \\ Rt \end{array} \begin{array}{c} \begin{array}{ccccc} WS_0 & WS_1 & WS_2 & WS_3 & WS_4 \end{array} \\ \begin{pmatrix} VG & G & G & F & F \\ VG & VG & VG & VF & L \\ C & C & E & M & M \\ L & VH & L & H & M \end{pmatrix} \end{array}$$

Classement avec l'algorithme TOPSIS flou

Avec l'application de l'algorithme TOPSIS flou :

— Matrice de décision pondérée

	WS_0	WS_1	WS_2	WS_3	WS_4
Rel	[0.6 0.8 3]	[0.35 0.5 2.1]	[0.6 0.8 3]	[0.15 0.3 1.5]	[0.026 0.026 0.026]
Av	[0.6 0.8 3]	[3 5.6 9]	[3 5.6 9]	[3 5.6 9]	[0 0 2.25]
pr	[3 5.6 9]	[0.25 5 9]	[0.133 0.952 3]	[0.25 5 9]	[0.25 5 9]
Rt	[0.016 0.03 0.06]	[0.009 0.014 0.018]	[0.031 0.9 9]	[0.012 0.02 0.026]	[0.016 0.030 0.06]

TABLE 7.10 – Matrice de décision pondérée

— Classement des services Web

	WS_0	WS_1	WS_2	WS_3	WS_4
d_i^-	11,355	10,771	12,420	10,411	7,782
d_i^*	20,381	20,689	20,571	20,944	25,004
CC_i	0,358	0,342	**0,376**	0,332	0,237

TABLE 7.11 – Scores attribués aux services Web

En appliquant l'algorithme MCDM conventionnel, on peut obtenir le classement des services Web suivant :

Service Web	**Score**
WS_0	0,56314
WS_1	0,27365
WS_2	**0,77358**
WS_3	0,34689
WS_4	0,57970

TABLE 7.12 – Classement des services Web

7.4 Résultats

Cette approche est testée pour des services Web similaires à l'aide de leurs critères de qualités : Fiabilité, disponibilité, prix et Temps de réponsee. Ces services Web sont classés selon l'algorithme Fuzzy TOPSIS comme dans le Tableau 7.12. Elle montre les valeurs de QoS initiales pour chaque service Web, puis calculer la matrice intermédiare floue qui sera utilisée plus tard pour calculer la matrice normalisée, la matrice normalisée pondérée et la distance de chaque service Web par rapport au FNIS et au FPIS.

Pour montrer la faisabilité de notre approche, nous avons testé ces services Web similaires avec deux méthodes de base : la méthode linéaire et aléatoire et une méthode, une méthode MCDM[95] et finalement avec notre proposition.

	Méthode linéaire	**méthode aléatoire**	**méthode MCDM Conventionnelle**	**Notre Approche**
Classement des services	WS_4	WS_3	WS_2	WS_2
Nombre de critères pris dans la sélection	1	Les critères ne sont pris en compte	Tous les critères	Tous les critères
Agents	Non	Non	Non	Oui
Type de données	Non flou	Non flou	Non flou	Flou

TABLE 7.13 – Tableau comparatif

Nous pouvons conclure que notre approche offrait le meilleur choix car d'une part, elle prend en compte tous les critères au cours du processus de sélection et d'autre part elle prend aussi les données linguistiques du consommateur du service ainsi que l'utilisation de l'agent paradigme dans notre approche peut apporter un gain très intéressant surtout en termes de temps de réponse.

Comme, nous voyons la dernière colonne du tableau précedent, le meilleur service est celui qui a le plus grand score donc le service WS_2 est le service classé alors il sera retourné à l'utilisateur.

Chapitre 8

Conclusion Générale et Perspectives

8.1 Conclusion

L'objectif de cette thèse est de proposer des solutions pour la problématique qui se résume en la prise en charge des données linguistiques(termes flous) lors de la sélection d'un service Web car les utilisateurs du Web sont en confrontation permanente entre ce qu'ils veulent et ce qu'ils reçoivent en conséquence dans un processus de découverte de service Web. L'utilisation des QoS dans le processus de découverte est un pas prometteur pour résoudre ce genre de problème mais comme, il est difficile à définir avec précision les valeurs des paramètres des QoS car les qualités des services liés aux consommateurs de services Web sont imprécises et parfois incertaines et ambiguës en raison de l'état mental et du manque d'information des consommateurs sur le contenu des services Web et aussi sur leur QoS, nous avons opté de profiter de la logique floue pour présenter les données linguistiques imprécises et incertaines.

Comme solution, nous avons présenté dans cette thèse deux approches de sélection des services Web, la première repose sur l'inférence floue alors que la deuxième se base en plus de la logique floue sur l'aspect agent.

Dans cette thèse, nous avons fait en premier lieu, un survol sur les technologies des services Web dont nous avons présenté l'état de l'art sur les services Web, les différentes architectures et notamment l'architecture en couches. En deuxième lieu, nous avons présenté la logique floue dont nous exposé plusieurs points à savoir : domaines d'applications de la logique floue, ensembles flous, fonctions d'appartenance, systèmes d'inférence flou enfin, nous terminons cette

section avec l'apport de la logique floue aux services Web. Et comme l'aspect agent et dans le cœur de notre solution, nous avons présenté quelques notions de cet aspect.

Nous avons terminé cette partie avec la présentation de quelques travaux reliés avec notre recherche afin de connaitre la technologie existante.

Dans la deuxième partie et à fin de résoudre le problème posée, nous avons exposé notre solution dans laquelle, nous proposons deux contributions :

La première contribution consiste en la proposition d'une modèle basé logique floue pour la sélection des services Web. Cette contribution a été enrichie avec un algorithme de sélection des services Web qui décrit le processus de sélection floue, dès l'émission de la requête par l'utilisateur, passant le sous-système flou qui s'occupe de la transformation floue et de l'inférence jusqu'à le retour du meilleur service au demandeur.

La deuxième contribution, est comme les agents logiciels sont considérés comme l'un des principaux éléments constituant l'infrastructure Web de la prochaine génération, nous nous sommes concentrés sur cette notion au cours du processus de sélection. Les services Web sont des ressources très importantes pour les agents. Les agents doivent être en mesure de récupérer, d'exécuter et de composer des services Web, offrant un support intelligent et personnalisé aux utilisateurs.

Cette approche est constituée principalement d'une couche basée agent jouant le rôle d'un système flou qui s'occupe de la transformation des critères de qualités de service en valeurs floues, formant une matrice de décision floue qui sera transmise à l'agent de clasement flou qui ensuite exécute l'algorithme TOPSIS floue pour classer ces services Web similaires.

Afin de montrer la faisabilité de notre solution, nous avons présenté dans le chapitre avant dernier une expérimentation numérique.

8.2 Perspectives

Dans le futur et comme perspectives de ce travail, nous considérerons les extensions suivantes :

— Intervenir des experts pour la définition d'une base de règles floues solide et réutilisable pour tester le modèle proposé ;

— Afin d'améliorer le processus de sélection, nous introduirons d'autres mesures de qualité de service ;

— Adapter ces modèles à la technologie de cloud computing ;

— Expérimenter notre approche avec une large dataset des services Web.

Annexe A

Liste des publications

A.1 Revues

Houcine BELOUAAR, Okba KAZAR, Nadia KABACHI, "A new model for web services selection based on fuzzy logic". Courrier du Savoir – N°26, Mars 2018, pp393-400.).

Houcine BELOUAAR, Okba KAZAR, " Agent and fuzzy logic based approach for Web service selection", Journal of Universal computer science (J.UCS), Vol. X, No. Y, pp.xxx–xxx. (**Soumis**).

Houcine BELOUAAR, Okba KAZAR, Nadia KABACHI, "Web service selection based on agent and fuzzy logic. "Iranian Journal of Mathematical Sciences and Informatics (IJMSI), Vol. X, No. Y, pp.xxx–xxx. (**Soumis**).

A.2 Conférences Internationales

Houcine BELOUAAR, Okba KAZAR, "Un modèle d'agent à raisonnement qualitatif pour le Web", Colloque International ISKO Maghreb' 2011,"Concepts and Tools for Knowledge Management", 13 et 14 mai 2011, Hammamet (Tunisie).

Houcine BELOUAAR, Okba KAZAR, Khaled Rezeg, "Web service selection based on TOPSIS algorithm", International Conference on Mathematics and Information Technology (ICMIT'

2017), 4-6 décembre 2017, Université d'Adrar-Algérie, pp. 173-184.

Houcine BELOUAAR, Okba KAZAR, Nadia KABACHI , "Agent based approach for Web service selection", International Conference on modern intelligent Systems Concepts (MISC'2018), December 12 - 13, 2018, Mohammed V University, Rabat- Morocco (**Accepté**).

Zouai Meftah, Kazar Okba, Haba Balgecem, Merizig Abdelhak and Belouaar Houcine, " A New Smart house design based on Multi-agent system and Raspberry PI card",Artificial Intelligence and its Applications (AIAP'2018), El Oued, Algeria, 04-05 December 2018 (**Accepté**).

Bibliographie

[1] Sheng, Quan Z., et al. "Web services composition : A decade's overview." Information Sciences 280 (2014) : 218-238.

[2] Wang, Hongbing, et al. "Web services : problems and future directions." Web Semantics : Science, Services and Agents on the World Wide Web 1.3 (2004) : 309-320.

[3] Kellert, Patrick, and Farouk Toumani. "Les web services sémantiques." Web sémantique, Action spéci_que 32 (2003).

[4] Kayastha, Ronak R., and Jwalant Baria. "A survey on web service selection and ranking methods." International Journal of Computer Applications 117.16 (2015).

[5] Tapang, Carlos C. "Web Services Description Language (WSDL) Explained." Microsoft Developer Network (2001).

[6] Negi, Neerja, and Satish Chandra. "Web service selection on the basis of QoS parameter." Contemporary Computing (IC3), 2014 Seventh International Conference on. IEEE, 2014.

[7] Guidi, Davide, Mauro Gaspari, and Giuseppe Profiti. "Web Services Integration in Multi-Agent Systems." Developing Advanced Web Services through P2P Computing and Autonomous Agents : Trends and Innovations : Trends and Innovations (2010) : 1.

[8] Antunes, João, André Vasconcelos, and José M. Tribolet. "Fuzzy Logic based Quality of Service Models." IJCCI (ECTA-FCTA). 2011.

[9] Schmidt, Stefan, et al. "Fuzzy service quality review in service oriented architectures." Fuzzy Systems, 2006 IEEE International Conference on. IEEE, 2006.

[10] Chakhar, Salem, et al. "Multicriteria evaluation-based conceptual framework for composite Web service selection." Lamsade, University Paris Dauphine, France, Research Report (2011).

[11] Mukhopadhyay, Debajyoti, and Archana Chougule. "A survey on web service discovery approaches." Advances in Computer Science, Engineering & Applications. Springer, Berlin, Heidelberg, 2012. 1001-1012.

[12] Bozkurt, Mustafa, Mark Harman, and Youssef Hassoun. "Testing web services : A survey." Department of Computer Science, King's College London, Tech. Rep. TR-10-01 (2010).

[13] http ://www.service-architecture.com/

[14] Dodani, Mahesh H. "From Objects to Services : A Journey in Search of Component Reuse Nirvana." Journal of Object Technology 3.8 (2004) : 49-54.

[15] Verjus, Hervé, and Frédéric Pourraz. "A formal framework for building, checking and evolving service oriented architectures." Web Services, 2007. ECOWS'07. Fifth European Conference on. IEEE, 2007.

[16] Zuo, Wei. Managing and modeling web service evolution in SOA architecture. Diss. Université de Lyon, 2016.

[17] Nahak, Sunil Kumar, Durga Prasad Mohapatra, and Manas Ranjan Patra. "Survey on Service Oriented Architecture Testing." IJACTA 2.2 (2015) : 102-109.

[18] Omrana, Hajar. Vers une composition dynamique des Services Web : une approche de composabilité offline. Diss. UNIVERSITÉ MOHAMMED V AGDAL–ECOLE MOHAMMADIA D'INGENIEURS, 2014.

[19] Lemos, Angel Lagares, Florian Daniel, and Boualem Benatallah. "Web service composition : a survey of techniques and tools." ACM Computing Surveys (CSUR) 48.3 (2016) : 33.

[20] Aljazzaf, Zainab. "Bootstrapping quality of web services." Journal of King Saud University-Computer and Information Sciences 27.3 (2015) : 323-333.

[21] Dustdar, Schahram, and Wolfgang Schreiner. "A survey on web services composition." International journal of web and grid services 1.1 (2005) : 1-30.

[22] Mathkour, Hassan, Sofien Gannouni, and Mutaz Beraka. "Web service composition : Models and approaches." Multimedia Computing and Systems (ICMCS), 2012 International Conference on. IEEE, 2012.

[23] Kreger, Heather. "Web services conceptual architecture (WSCA 1.0)." IBM software group 5.1 (2001) : 6-7.

[24] Gottschalk, Karl, et al. "Introduction to web services architecture." IBM systems Journal 41.2 (2002) : 170-177.

[25] Leutenmayr, Stephan. "Selected Languages for Web Services Composition : Survey, Challenges, Outlook." (2007).

[26] Chinnici, Roberto, et al. "Web services description language (WSDL) version 2.0 part 1 : Core language." W3C working draft 26 (2004).

[27] Hadjila, FethAllah. Composition et interopération des services web sémantiques. Diss. 2014.

[28] Charfi, Anis, et al. "An overview of the unified service description language." Web Services (ECOWS), 2010 IEEE 8th European Conference on. IEEE, 2010.

[29] Barton, Lukas, et al. "Unified Service Description Language XG Final Report." (2011).

[30] Zhang, Chuanrong, Tian Zhao, and Weidong Li. Geospatial semantic web. Springer, 2015.

[31] Nacer, Hassina, and Djamil Aissani. "Semantic web services : Standards, applications, challenges and solutions." Journal of Network and Computer Applications 44 (2014) : 134-151.

[32] Gustavo, Alonso, et al. "Web services : concepts, architectures and applications." (2004).

[33] Merizig, Abdelhak. Approche de composition de services web dans le Cloud Computing basée sur la coopération des agents. Diss. Université Mohamed Khider-Biskra, 2018.

[34] Webber, J., and S. Chatterjee. "Developing Enterprise Web Services : An Architect's Guide." (2003).

[35] Roman, Dumitru, et al. "Web service modeling ontology." Applied ontology 1.1 (2005) : 77-106.

[36] Akkiraju, Rama, et al. "Web service semantics-wsdl-s." (2005).

[37] M'Barek, Nomane Ould Ahmed, and Samir Tata. "Services Web : revue des approches de description sémantique." SIIE 2008 : Systéme dÍnformation et Intelligence Economique. 2008.

[38] Lopez-Velasco, Céline. Sélection et composition de services Web pour la génération d'applications adaptées au contexte d'utilisation. Diss. Université Joseph-Fourier-Grenoble I, 2008.

[39] Nadalin, Anthony, et al. "Web services security : SOAP message security 1.1 (WS-Security 2004)." Oasis Standard 200401 (2006).

[40] Sheng, Quan Z., et al. "Self-serv : A platform for rapid composition of web services in a peer-to-peer environment." VLDB'02 : Proceedings of the 28th International Conference on Very Large Databases. 2002.

[41] Benatallah, Boualem, Marlon Dumas, and Quan Z. Sheng. "Facilitating the rapid development and scalable orchestration of composite web services." Distributed and Parallel Databases 17.1 (2005) : 5-37.

[42] Medjahed, Brahim. Semantic web enabled composition of web services. Diss. Virginia Tech, 2004.

[43] Benatallah, Boualem, Marlon Dumas, and Zakaria Maamar. "Definition and execution of composite web services : The self-serv project." IEEE Data Eng. Bull. 25.4 (2002) : 47-52.

[44] Lee, Jonathan, et al. "Dynamic service composition : A discovery-based approach." International Journal of Software Engineering and Knowledge Engineering 18.02 (2008) : 199-222.

[45] HALFAOUI épouse GHERNAOUT, Amal. La sélection des services web dans une composition à base de critères non fonctionnels. Diss. 19/03/2017, 2017.

[46] Yanping, Chen, et al. "Study on life cycle model of dynamic composed web services." International Conference on Algorithms and Architectures for Parallel Processing. Springer, Berlin, Heidelberg, 2005.

[47] Sivanandam, S. N., Sai Sumathi, and S. N. Deepa. Introduction to fuzzy logic using MATLAB. Vol. 1. Berlin : Springer, 2007.

[48] Dernoncourt, Franck, and Elisabeth Métais. "La Logique Floue : le raisonnement humain au cœur du systeme décisionnel?." Memory NFE211 engineering decision systems Paris (February 2011) http ://www. academia. edu/1053161 (2011).

[49] Houcine BELOUAAR, Okba KAZAR, Nadia KABACHI, "A new model for web services selection based on fuzzy logic". Courrier du Savoir – N°26, Mars 2018, pp393-400.).

[50] Hema Priya, N., and S. Chandramathi. "QoS Based Optimal Selection of Web Services Using Fuzzy Logic." Journal of Emerging Technologies in Web Intelligence 6.3 (2014).

[51] Malik, Mohamed Mahdi. "Le rôle de la logique floue dans le web sémantique." Secrétariat du DEA d'Informatique—Université de la Méditerranée Faculté des Sciences de Luminy—Case 901 163, Avenue de Luminy—13288 MARSEILLE Cedex 9 Tel : 0 491 829 316—Télécopie : 0 491 829 275 secrétariat : secretariat-dea@ dil. univ-mrs. fr (2002) : 263.

[52] Omar, Amina S., Mwangi Waweru, and Richard Rimiru. "A Literature Survey : Fuzzy Logic and Qualitative Performance Evaluation of Supply Chain Management." The International Journal Of Engineering And Science (IJES) Volume 4 : 56-63.

[53] Zadeh, Lotfi A. "Is there a need for fuzzy logic?." Information sciences 178.13 (2008) : 2751-2779.

[54] Zadeh, Lotfi Asker. "Fuzzy logic." Computer 21.4 (1988) : 83-93.

[55] Zadeh, Lotfi A. "Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic." Fuzzy sets and systems 90.2 (1997) : 111-127.

[56] Leonid, Reznik. "Fuzzy controllers." Newnes 1 (1997) : 1-9.

[57] Lin, Chi-Jen, and Wei-Wen Wu. "A causal analytical method for group decision-making under fuzzy environment." Expert Systems with Applications 34.1 (2008) : 205-213.

[58] Bai, Ying, and Dali Wang. "Fundamentals of fuzzy logic control—fuzzy sets, fuzzy rules and defuzzifications." Advanced Fuzzy Logic Technologies in Industrial Applications. Springer, London, 2006. 17-36.

[59] Banks, Walter. "Linguistic variables : Clear thinking with fuzzy logic." IEEE Toronto Section (2008).

[60] Wierzchon, Slawomir T. "The fuzzy systems handbook. A practitioner's guide to building, using, and maintaining fuzzy systems : by Earl COX; AP Professional; Boston, MA, USA; 1994; xxxix+ 624 pp.; $49–95; ISBN : 0-12-194270-8." (1995) : 1352-1353.

[61] Chevrie, François, and François Guély. "La logique floue." Cahier technique 191 (1998).

[62] Ambapour, Samuel. "Théorie des ensembles flous : application à la mesure de la pauvreté au Congo." DT 16 (2009) : 2009.

[63] Ayouni, Sarra. "Etude et extraction de regles graduelles floues : définition d'algorithmes efficaces." These de doctorat, Université Montpellier 2 (2012).

[64] Kauffman, A., and Madan M. Gupta. "Introduction to fuzzy arithmetic : theory and application." (1991).

[65] Tran, Vuong Xuan, and Hidekazu Tsuji. "QoS based ranking for web services : Fuzzy approaches." Next Generation Web Services Practices, 2008. NWESP'08. 4th International Conference on. Ieee, 2008.

[66] Liu, Yutu, Anne H. Ngu, and Liang Z. Zeng. "QoS computation and policing in dynamic web service selection." Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters. ACM, 2004.

[67] Buvanesvari, R., V. Prasath, and H. SanofarNisha. "A review of fuzzy based QoS web service discovery." International Journal of Advanced Networking and Applications 4.5 (2013) : 1752.

[68] Kuyoro Shade, O., et al. "Quality of service (Qos) issues in web services." IJCSNS International Journal of Computer Science and Network Security 12.1 (2012) : 94-97.

[69] Menascé, Daniel A. "QoS issues in web services." IEEE internet computing 6.6 (2002) : 72-75.

[70] Zheng, Zibin, and Michael R. Lyu. "Qos evaluation of web services." QoS Management of Web Services. Springer, Berlin, Heidelberg, 2013. 19-39.

[71] Mani, Anbazhagan, and Arun Nagarajan. "Understanding quality of service for Web services." (2005).

[72] Karim, Raed, Chen Ding, and Ali Miri. "An end-to-end QoS mapping approach for cloud service selection." Services (SERVICES), 2013 IEEE Ninth World Congress on. IEEE, 2013.

[73] Shafiq, M. Omair, Ying Ding, and Dieter Fensel. "Bridging multi agent systems and web services : towards interoperability between software agents and semantic web services." Enterprise Distributed Object Computing Conference, 2006. EDOC'06. 10th IEEE International. IEEE, 2006.

[74] Guériau, Maxime. Systèmes multi-agents, auto-organisation et contrôle par apprentissage constructiviste pour la modélisation et la régulation dans les systèmes coopératifs de trafic. Diss. Université de Lyon I Claude Bernard, 2016.

[75] Jennings, Nicholas R. "On agent-based software engineering." Artificial intelligence 117.2 (2000) : 277-296.

[76] Biswas, Pratik K. "Towards an agent-oriented approach to conceptualization." Applied Soft Computing 8.1 (2008) : 127-139.

[77] Triantaphyllou, Evangelos, et al. "Multi-criteria decision making : an operations research approach." Encyclopedia of electrical and electronics engineering 15.1998 (1998) : 175-186.

[78] Rajeswari, M., et al. "Appraisal and analysis on various web service composition approaches based on QoS factors." Journal of King Saud University-Computer and Information Sciences 26.1 (2014) : 143-152.

[79] Nădăban, Sorin, Simona Dzitac, and Ioan Dzitac. "Fuzzy topsis : A general view." Procedia Computer Science 91 (2016) : 823-831.

[80] Wang, Ying-Ming, and Taha MS Elhag. "Fuzzy TOPSIS method based on alpha level sets with an application to bridge risk assessment." Expert systems with applications 31.2 (2006) : 309-319.

[81] Madi, Elissa Nadia, Jonathan M. Garibaldi, and Christian Wagner. "A comparison between two types of Fuzzy TOPSIS Method." Systems, Man, and Cybernetics (SMC), 2015 IEEE International Conference on. IEEE, 2015.

[82] Demazeau, Yves, and J-P. Müller, eds. Decentralized Ai. Vol. 2. Elsevier, 1990.

[83] Frécon, Louis, and Okba Kazar. Manuel d'intelligence artificielle. PPUR Presses polytechniques, 2009.

[84] Perret, Stéphane. Agents mobiles pour l'accès nomade à l'information répartie dans les réseaux de grande envergure. Diss. Grenoble 1, 1997.

[85] Ferber, Jacques. Les systèmes multi-agents : vers une intelligence collective. InterEditions, 1997.

[86] Wooldridge, Michael, and Nicholas R. Jennings. "Intelligent agents : Theory and practice." The knowledge engineering review 10.2 (1995) : 115-152.

[87] Jennings, Nicholas R., Katia Sycara, and Michael Wooldridge. "A roadmap of agent research and development." Autonomous agents and multi-agent systems 1.1 (1998) : 7-38.

[88] Labidi, Sofiane, and Wided Lejouad. De l'intelligence artificielle distribuée aux systèmes multi-agents. Diss. INRIA, 1993.

[89] Kabachi, Nadia. Modélisation et apprentissage de la prise de décision dans les organisations productives : approche multi-agents. Diss. Ecole Nationale Supérieure des Mines de Saint-Etienne ; Université Jean Monnet-Saint-Etienne, 1999.

[90] Hamida, Soraya. Une approche basée agent mobile pour le m-service web sémantique. Diss. Université Mohamed Khider Biskra, 2014.

[91] Hayes-Roth, Barbara. "A blackboard architecture for control." Readings in Distributed Artificial Intelligence. 1988. 505-540.

[92] Huhns, Michael N. "Agents as Web services." IEEE Internet computing 6.4 (2002) : 93-95.

[93] Seghir, Nadia Ben, Okba Kazar, and Khaled Rezeg. "A decentralized framework for semantic web services discovery using mobile agent." International Journal of Information Technology and Web Engineering (IJITWE) 10.4 (2015) : 20-43.

[94] Zou, Guobing, et al. "An agent-based web service selection and ranking framework with QoS." Computer Science and Information Technology, 2009. ICCSIT 2009. 2nd IEEE International Conference on. IEEE, 2009.

[95] Belouaar, Houcine, Okba Kazar, and Khaled Rezeg. "Web service selection based on TOPSIS algorithm." Mathematics and Information Technology (ICMIT), 2017 International Conference on. IEEE, 2017.

[96] Gohar, Priya, and Lalit Purohit. "Discovery and prioritization of web services based on fuzzy user preferences for QoS." Computer, Communication and Control (IC4), 2015 International Conference on. IEEE, 2015.

[97] Gharbi, Atef, and B. A. Samir. "Fuzzy logic multi-agent system." Int. J. Comput. Sci. Inf. Technol 6.4 (2014) : 273.

[98] Awasthi, Anjali, Satyaveer S. Chauhan, and Surseh K. Goyal. "A fuzzy multicriteria approach for evaluating environmental performance of suppliers." International Journal of Production Economics 126.2 (2010) : 370-378.

[99] Cui, Zhu Xian, et al. "Multi-criteria group decision making with fuzzy logic and entropy based weighting." Proceedings of the 5th International Conference on Ubiquitous Information Management and Communication. ACM, 2011.

[100] Keller, Uwe, et al. "Wsmo web service discovery." WSML Working Draft D 5 (2004).

[101] Mohebbi, Keyvan, et al. "A comparative evaluation of semantic web service discovery approaches." Proceedings of the 12th International Conference on Information Integration and Web-based Applications Services. ACM, 2010

[102] Hwang, Ching-Lai, and Kwangsun Yoon. "Multiple criteria decision making." Lecture Notes in Economics and Mathematical Systems 186 (1981) : 58-191.

[103] Roy, B. "Classement et choix en présence de points de vue multiples (la méthode ELECTRE), Rev. Française Automat., Informat." Recherche Opérationnelle" (1968).

[104] Kousalya, P., et al. "Analytical Hierarchy Process approach–An application of engineering education." Mathematica Aeterna 2.10 (2012) : 861-878.

[105] Al-Masri, Eyhab, and Qusay H. Mahmoud. "Qos-based discovery and ranking of web services." Computer Communications and Networks, 2007. ICCCN 2007. Proceedings of 16th International Conference on. IEEE, 2007.

[106] Chen, Vivien YC, et al. "Fuzzy MCDM approach for selecting the best environment-watershed plan." Applied soft computing 11.1 (2011) : 265-275.

[107] Garg, Harish, Nikunj Agarwal, and A. Tripathi. "Entropy based multi-criteria decision making method under fuzzy environment and unknown attribute weights." Glob J Technol Optim 6.3 (2015) : 13-20.

[108] Li, Yifan, Petr Musilek, and Loren Wyard-Scott. "Fuzzy logic in agent-based game design." Proceedings of the 2004 annual meeting of the North American fuzzy information processing society. 2004.

[109] unika, Wlodzimierz, Filip Szura, and Jacek Kitowski. "Agent-based monitoring using fuzzy logic and rules." Computer Science 12 (2011) : 103-113.

[110] Ropero, Jorge, et al. "A Fuzzy Logic intelligent agent for Information Extraction : Introducing a new Fuzzy Logic-based term weighting scheme." Expert Systems with Applications 39.4 (2012) : 4567-4581

[111] https ://www.ibm.com/support/knowledgecenter/en/SSGMCP_5.3.0/com.ibm.cics.ts.webservices.doc/concepts/dfhws\definition.html.

[112] https ://www.w3.org/TR/2001/NOTE-wsdl-20010315#_introduction

[113] https ://www.w3.org/TR/soap12/

[114] https ://www.figer.com/Publications/SOA.htm

[115] https ://openclassrooms.com/courses/les-services-web

[116] http ://docs.embarcadero.com/products/rad\studio/radstudio2007/RS2007\helpupdates/HUpdate4/FR/html/devnet/webservicesprotocol_\xml.htm

[117] https ://openclassrooms.com/courses/les-services-web

[118] http ://www.computerworld.com/news/2004/story/0,11280,95282,00.html

[119] https ://en.wikipedia.org/wiki/Fuzzy_logic 11/04/2017

[120] http ://www.ferdinandpiette.com/blog/2011/08/les-systemes-flous-le-fonctionnement/